

Joonas Suni

# Kappaletavaralinjaston simulointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinöörityö

9.11.2013

Tekijä(t) Otsikko	Joonas Suni Kappaletavaralinjaston simulointi
Sivumäärä Aika	27 sivua + 2 liitettä 9.11.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	automaatiotekniikka
Suuntautumisvaihtoehto	kappaletavara-automaatio
Ohjaaja(t)	lehtori Timo Tuominen yliopettaja Jouni Jokelainen
<p>Tämän insinöörityön aiheena oli selvittää simulointiprojektien ja yleisesti simuloinnin teoriaa, sekä toteuttaa Delmia V5 -ohjelmiston avulla virtuaalinen kappaletavaralinjaston malli. Työ tehtiin Metropolia Ammattikorkeakoulun automaatiotekniikan koulutusohjelman toimeksiannosta, ja siihen liittyy myös kaksi samaan aikaan suoritettavaa insinöörityötä. Yhdessä näistä kolmesta työstä muodostui yksi projekti, jonka tarkoituksena oli tuottaa koululle opetuskäyttöön kokonainen simuloitu tuotantojärjestelmä.</p> <p>Insinöörityössä tutustuttiin Delmia V5 -ohjelmiston tarjoamiin mahdollisuuksiin simulointimalleja luotaessa, niiden ohjaamisessa, sekä niiden liittämismahdollisuuksiin ulkoisiin logiikkaohjelmiin. Yleisesti selvitettiin erilaisia tapoja tuotannon tehostamiseen sekä ongelmien ratkaisemiseen. Perehdyttiin myös mahdollisiin ongelmiin ja haasteisiin simuloitaessa tuotantojärjestelmiä.</p> <p>Kappaletavaralinjasto toteutettiin piirtämällä yksinkertainen 3D-malli. Työssä perehdyttiin 3D-mallien luomiseen valmistusjärjestelmien simulointikäyttöön, sekä yleisesti tietokoneavusteiseen mallintamisen teoriaan.</p> <p>Työn lopputuloksena saatiin toimiva simulaatiomalli kappaletavaralinjaston osasta, jota pystytään tarvittaessa ohjaamaan ulkopuolisella logiikkaohjelmalla. Tehtyä mallia voidaan käyttää tulevaisuudessa tuotantojärjestelmän simuloinnissa. Työssä tuotettiin myös automaatiotekniikan koulutusohjelman käyttöön ohjeet sekä tekopohjat kahdelle erilliselle laboratoriotyölle Delmia V5 -ohjelmalla.</p>	
Avainsanat	Simulointi, Delmia, 3D-malli, OPC

Author(s) Title	Joonas Suni Simulation of a production system
Number of Pages Date	27 pages + 2 appendices 9.11.2013
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialisation option	Manufacturing Automation
Instructor(s)	Timo Tuominen, Senior Lecturer Jouni Jokelainen, Principal Lecturer
<p>The purpose of this thesis was to find out about the theory of simulation projects and simulating in general and to create a virtual model of a production line with Delmia V5 software. This thesis was commissioned by the Degree Programme of Automation Engineering at the Helsinki Metropolia University of Applied Sciences, and it was one of three theses being carried out at the same time. Together these three theses created one project, which aimed to produce an entire simulated production system for the school's educational use.</p> <p>The thesis explored the possibilities offered by Delmia V5 software in creating, controlling and connecting the simulation models into external logic programs. In general, the plan was to research different ways to enhance productivity and to solve problems. The aim was also to become familiar with possible problems and challenges involved in creating simulated production systems.</p> <p>The production line of this project was carried out by drawing a simple 3D model. Creating 3D models for the use of manufacturing systems as well as general computer-aided modeling theory were examined.</p> <p>As a result of the project, a functional part of the simulation model of the production line was achieved, which is able to be controlled by an external logic program if needed. The model created can be used in the future to simulate a production system. Also instructions for the use of Delmia V5 software and two sets of new laboratory exercises were created.</p>	
Keywords	simulation, Delmia, 3D model, OPC

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Projektin kuvaus	2
2.1	Alkutilanne	2
2.2	Kuvaus systeemistä	2
2.3	Projektin tavoitteet ja suunnitellut vaiheet	3
2.4	Toimintasuunnitelma	4
3	Delmia	5
3.1	Dassault Systèmes S.A.	5
3.2	Delmia Automation	5
4	Simulointi	6
4.1	Määritelmiä	6
4.1.1	CAD	6
4.1.2	CAE	7
4.1.3	PLM	7
4.1.4	Bottom-up-suunnittelu	7
4.1.5	Top-down-suunnittelu	7
4.1.6	OPC	7
4.2	Reaaliaikainen simulointi	8
4.2.1	Synkronoitu malli	8
4.2.2	Automaattisesti rakennettava malli	9
4.3	Simulointiprojektin vaiheet	10
4.3.1	Yleistä	10
4.3.2	Mallin suunnittelu	11
4.3.3	Mallin kehittäminen	12
4.3.4	Mallin käyttöönotto	12
4.3.5	Ongelmia	13
4.3.6	Hyötyjä	14
5	Toteutus	16
5.1	Osien luominen	16
5.2	Kokoonpano	17

5.3	Mekanismit	18
5.4	Anturit	20
5.5	OPC	21
5.6	Simulointi	23
6	Yhteenveto	25
6.1	Aikaansaatu kokonaisuus ja vertailu tavoitteisiin	25
6.2	Kehitysmahdollisuudet	26
6.3	Laboratoriotyöt	26
	Lähteet	28
	Liitteet	
	Liite 1. OPC Delmia -laboratorioharjoitus	
	Liite 2. Delmia Process -laboratorioharjoitus	

## Lyhenteet

3D	Three-dimensional. Kolmiulotteinen.
Assembly	Koonta, joka muodostuu useamman osan yhteenliittämisestä.
CAD	Computer-aided Design. Tietokoneavusteinen suunnittelu.
CAE	Computer-aided Engineering. Tietokoneavusteinen insinöörityö.
Catia	Conception Assistée Tridimensionnelle Interactive Appliquée. Dassault Systèmesin kehittämä 3D-mallinnusohjelma.
Delmia	Dassault Systèmesin kehittämä digitaalinen valmistuksen ja tuotannon ohjelmistoratkaisu.
OPC	Open Connectivity via Open Standards. Ohjelmointirajapinta standardi.
Part	Yksittäinen kappale, eli osa 3D-ohjelmassa.
PLM	Product Lifecycle Management. Tuotteen elinkaaren hallinta.
Sketch	3D-mallinnuksessa määrittelee kappaleen rajat.

## 1 Johdanto

Tämä opinnäytetyö toteutetaan Metropolia Ammattikorkeakoulun Myyrmäen toimipisteessä sijaitsevassa automaatiolaboratoriossa. Opinnäytetyössä on tarkoituksena tehdä simulaatiomalli automaatiolaboratoriosta löytyvästä kappaleentuotantolinjastosta. Mallin luominen toteutetaan automaatiolaboratorioon valmiiksi asennetun Delmia V5 -ohjelmiston avulla.

Mallin tekemisessä ei pyritä absoluuttiseen tarkkuuteen, vaan sen tarkoituksena on olla enemmänkin suuntaa antava. Mallin tekemiseen pystyisi käyttämään huomattavasti aikaa, mutta juuri ajan säästämiseksi pyritään se pitämään kohtuullisen yksinkertaisena. Toisaalta mitä tarkempi malli on, sitä raskaampi se on tietokoneelle myös pyörittää.

Työ toteutetaan samanaikaisesti kahden muun opinnäytetyön kanssa, ja näistä on tarkoitus muodostaa kokonainen tuotantojärjestelmä. Simulointimalli on tarkoituksena liittää automaatiolaboratorion uuteen Siemens S7-1200 logiikkaan ja WinCC -valvomoympäristöön. Siemensin logiikka on puolestaan tarkoitus liittää Rob-Ex-tuotannonsuunnittelujärjestelmään. Tuotantojärjestelmällä on tarkoitus pystyä simuloimaan kokopäiväistä kappaleiden tuotantoa ja se voidaan jättää käyntiin keräämään tietoa valmistuksenohjausjärjestelmälle.

Tavoitteena on, että tuotantosuunnittelujärjestelmään syötetään tilaukset, jossa ne suunnitellaan, optimoidaan ja laitetaan eteenpäin tuotantoon. Tuotteet tuotetaan simuloitusti, ja tästä saadaan tietoa tuotannonsuunnittelujärjestelmään.

Tarkoituksena on myös Delmian käyttämiseen perehdyttävien laboratorioharjoitusten valmisteleminen myöhempää opetuskäyttöä varten.

## 2 Projektin kuvaus

### 2.1 Alkutilanne

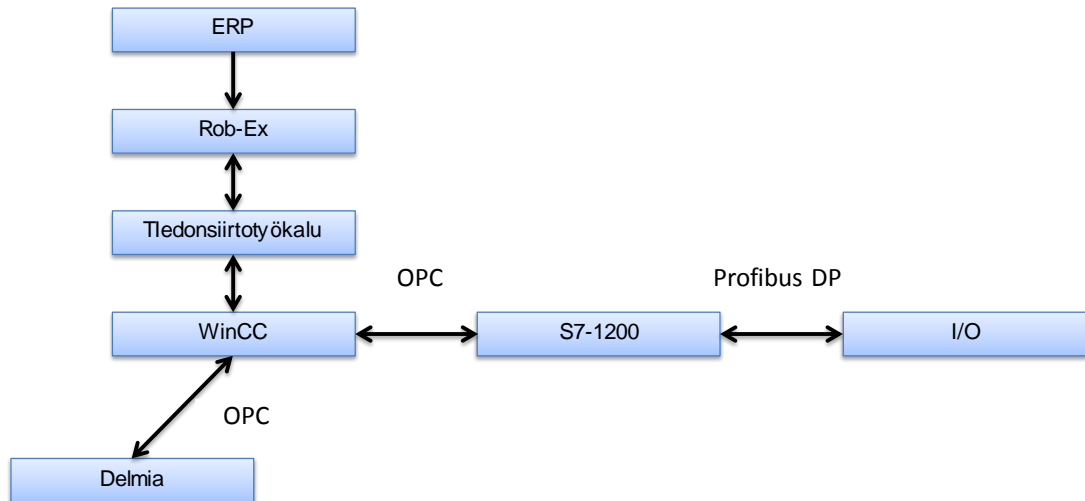
Opinnäytetyö pohjana käytetään Heikki Hietalan vuonna 2012 tekemää Simulointi Delmia V5R19 -ohjelmistolla -insinööriytöä. Työssään Hietala oli tehnyt linjastolla olevan prässikoneen ja tämän liitännän OPC:lla Beckhoff Twincat -logiikkaohjelmaan. Tämän ansiosta käytössä oli työtäni varten jo yksi linjaston osista ja malli OPC-rajapintaa varten. Delmia-ohjelmisto oli valmiina asennettuna yhdelle pc:lle, jossa Hietalan luoma malli oli valmiina ja toimintakunnossa. Tarkoituksena oli jatkaa linjaston tekemistä siitä mihin Hietala oli työssään jäänyt, sekä saada aikaan kokonaisvaltainen tuotantojärjestelmä, jolla voidaan simuloida tuotannonohjausta kokonaisen tuotantoprosessin avulla.

### 2.2 Kuvaus systeemistä

Metropolia Ammattikorkeakoulun automaatiolaboratorioon toteutettava tuotantojärjestelmä koostuu kolmesta eri opinnäytetyöstä. Oman osuuteni eli Delmiällä tehty 3D-mallin ja sen OPC-liitännän tekemisen lisäksi projektiin kuuluvan toisen työn tarkoituksena on kappaletavaratuotantolinjan PLC logiikan uudistaminen Siemensin S7-1200 logiikaksi, sekä valvomosovelluksen luominen WinCC-valvomo-ohjelmistolla. Kolmannen työn tarkoituksena taas on Rob-Ex-tuotannonsuunnittelujärjestelmän integrointi automaatiojärjestelmään, sekä tiedonsiirron luominen tuotannonsuunnittelujärjestelmän ja automaatiota ohjaavan valvomosovelluksen välille kuvan 1 mukaisesti.

Tarkoituksena on, että Rob-Ex-tuotannonsuunnittelujärjestelmään syötetään ja suunnitellaan tilaukset. Nämä tilaukset syötetään automaatiota ohjaavaan valvomoon, jossa tilaukset laitetaan tuotantoon. Delmia toteuttaa virtuaalisesti nämä tuotantoon laitettut tilaukset ja tuotannossa tapahtuneista ongelmista, kuten pysähdyksistä on mahdollista saada tietoa takaisinpäin tuotannonohjausjärjestelmään.





Kuva 1. Koko projektin kuvaus.

### 2.3 Projektin tavoitteet ja suunnitellut vaiheet

Yleisesti projektin tavoitteena on tuottaa koululle kokonainen tuotantojärjestelmä, jolla voidaan simuloida linjaston toimintaa ilman fyysisiä laitteistoja. Tuotantojärjestelmää voidaan myöhemmin käyttää opetuskäytössä ja laboratorioharjoituksissa.

Tämän osuuden tarkoituksena on tutkia yleisesti simuloimisen käyttämistä tuotantojärjestelmissä ja sen tarjoamia hyötyjä ja mahdollisuuksia. Tarkoituksena on myös saada mahdollisimman paljon tutkimustietoa Delmian käyttämisestä sekä sen ominaisuuksista ja mahdollisuuksista. Tarkoituksena on tehdä yhteistyötä muiden projektiin osallistuvien kanssa, jotta saadaan aikaan toimiva kokonaisuus.

Yhteistyötä eri osien välillä pyritään helpottamaan projektin aikana pitämällä säännöllisiä projektipalavereita, joissa seurataan edistymistä ja mietitään eri osien integrointia keskenään. Jokainen tekee myös alustavan aikataulun ja yleissuunnitelman projektin etenemisestä ja näitä seurataan palavereissa.

Projektini koostuu karkeasti neljästä eri osuudesta. Tutustuttuani ensin ohjelmistoon ja sen tarjoamiin mahdollisuuksiin aloitan linjaston osien piirtämisen linjaston toimittajan tarjoamien piirustusten pohjalta. Seuraava vaihe on osien mekanismien ja toiminnallisuuksien tekeminen. Kun osat liikkuvat halutulla tavalla lisätään OPC-rajapinta ohjel-

maan. Tämän työn tavoitteena ei ole sen suuremmin liittää mallia logiikkaohjelmistoon, vaan saada vain aikaiseksi OPC-rajapinta Delmiasta poispäin.

Dokumentointia projektin etenemisestä suoritetaan samalla työn tekemisen kanssa ja näin saadaan mahdollisimman selkeä yleiskuvaus projektista, sekä hyödyllistä tietoa esiintyvistä ongelmista ja näiden ratkaisuksista. Tällä dokumentoinnilla on myös tarkoitus helpottaa tulevaisuudessa mahdollista järjestelmän jatkokehitystä ja laajentamista.

## 2.4 Toimintasuunnitelma

Projekti aloitetaan tutustumalla Delmiaan, sen tajoamiin toiminnallisuuksiin ja ominaisuuksiin. Apuna ohjelmistoon tutustumisessa käytetään Heikki Hietalan tekemää präsiaseman simulointimallia ja sen OPC-liitäntää Beckhoff Twincat -logiikkaohjelmaan. Tämän valmiin mallin avulla on ohjelmistosta tarkoitus saada jonkinlainen yleiskuva ja hieman ymmärtää sen toimintaa ja simulointi käytössä hyödynnettäviä ominaisuuksia.

Kun ohjelmistoon on tutustuttu, tehdään suuntaa-antava aikataulu projektin kulusta aina työvaiheittain. Projektin edistymistä ja aikataulun noudattamista pyritään dokumentoimaan ja tarkkailemaan projektin kuluessa mahdollisuuksien mukaan, näin projekti pyritään saamaan tavoitteeseen mennessä valmiiksi. Tällä aikataulutuksella pyritään myös helpottamaan yhteistyötä projektin eri osien välillä, kun eri osien pitää toimia keskenään. Tehdään myös suullinen yhteissuunnitelma siitä missä vaiheessa eri osien on tarkoitus olla valmiina testattavaksi ja alustava suunnitelma siitä mitä tiedonsiirto-tekniikoita pyritään käyttämään.

Kun yhteissuunnitelmat ja oma aikataulu, sekä projektin päämäärät ovat selvillä, aloitetaan 3D-mallin osien piirtäminen valmiina olemassa olevien piirustusten pohjalta. Linjaston 3D-mallien osien valmistuttua lisätään niihin mekanismit, liikkeet ja tarvittavat yhteydet. Projektipalavereita on tarkoitus pitää kerran viikossa, näissä on tarkoitus toteuttaa edistymisen seuranta sekä käydä läpi ongelmia ja esittää kysymyksiä.

Loppuraporttia ja muuta dokumentointia on tarkoitus kirjoittaa samalla kun työtä tehdään. Näin projektista on helpompi muodostaa kokonaiskuva, ja kaikki ongelmat ja muut tärkeät asiat saadaan dokumentoitua. Kun simulointimalli on luotu ja saatu toimimaan halutulla tavalla, tehdään sen pohjalta vaaditut laboratorioharjoitukset automaatiotekniikan koulutusohjelman käyttöön.

### 3 Delmia

#### 3.1 Dassault Systèmes S.A.

Ranskalainen yhtiö Dassault Systèmes S.A. on SAP:in jälkeen Euroopan toiseksi suurin ohjelmistoyritys, ja sillä on yli 10 000 työntekijää noin 40 maassa. Vuonna 1981 perustettu yritys on markkinajohtaja tuotannon 3D-suunnittelun, digitaalisten 3D-mallien, sekä tuotteen elinkaaren hallinnan, eli PLM (Product Lifecycle Management) ratkaisuis-  
sa. Suomessa Dassault Systèmes Oy:n konttori sijaitsee Falcon Business Parkissa Espoossa Otaniemessä. Dassault Systèmesin tunnetuimpiin tuotteisiin kuuluvat muun muassa: Catia, SolidWorks, Delmia, Simulia ja DraftSight. [7;8]

#### 3.2 Delmia Automation

Delmia Automation on Dassault Systèmes S.A. -yhtiön digitaalinen valmistuksen ja tuotannon ohjelmistoratkaisu. Ohjelmistolla pystytään esimerkiksi suunnittelemaan prosesseja ja simuloimaan kokoonpanoja, mallintamaan tuotantoa, ohjelmoimaan robotteja ja soluja off-line ohjelmoinnilla. Suomessa Delmia suunnitteluratkaisuja edustaa Espoon Perkaalla sijaitseva Delfoi Oy. Delmia tuotemerkki perustettiin tammikuussa 2000, kun Deneb Robotics, EAI-Delta ja Safework yhdistettiin yhdeksi kokonaisuudeksi. [2;9]

Delmia-ohjelmistolla on mahdollista määritellä, suunnitella, luoda, seurata ja hallita lähestulkoon kaikkia tuotantoprosesseja toimialasta riippumatta. Tuotteen elinkaaren hallinta (PLM) auttaa valmistajia olemaan vuorovaikutuksessa ennen varsinaiseen tuotantoon sitoutumista, kuten myös tehtaan suunnitteluprosessin alussa. 3D-visualisoinnin avulla voidaan tunnistaa ja poistaa mahdollisia virheitä ja ongelmakohtia tuotannossa. Riskit pienenevät, kun tehtaiden tuotto ja toimivuus voidaan todeta simuloimalla etukäteen tarkasti. [2;9]

Kirjoitushetkellä uusin versio Delmian-ohjelmistosta on uusimmalla käyttöliittymällä varustettu V6R2013x. Koululla oleva Delmia V5R19 -ohjelmisto on edelliseen V5-käyttöliittymään perustuvan ohjelmiston yhdeksästoista julkaisuversio. [2;9]

## 4 Simulointi

Simulointi on ajan mittaan tapahtuvaa reaalimaailman prosessin tai järjestelmän jäljittämistä. Simulaatiosta voidaan kerätä dataa aivan kuten oikeistakin järjestelmistä. Tätä kerättyä dataa voidaan käyttää arvioimaan järjestelmän suorituskykyä. Simuloinnin avulla voidaan tuottaa teknisiä ja taloudellisia tunnuslukuja tuotekustannuslaskentaan, tuotannonohjaukseen ja investointien suunnitteluun. Graafisen kolmiulotteisen animaation avulla tulosten tulkitsemista voidaan joissakin tapauksissa nopeuttaa ja helpottaa, sillä se on helpommin ymmärrettävissä koulutuksesta ja työnkuvasta riippumatta. [1,2]

Simulaatiota voidaan käyttää joko suunnitteluvaiheessa, tai jo olemassa oleviin järjestelmiin, joihin halutaan tehdä muutoksia. Simulaatiomalleja voidaan käyttää yhtäläillä analyysityökaluina ennustamaan muutoksien vaikutusta jo olemassa oleviin järjestelmiin kuin suunnittelutyökaluna ennustamaan vielä rakentamattomien järjestelmien suorituskykyä erilaisissa tilanteissa. Simulointi voidaan toteuttaa joko käsin tai tietokoneella. Joissain yksinkertaisimmissa tapauksissa tarvittavien tulosten saamiseksi riittävät matemaattiset menetelmät, kuten todennäköisyyslaskenta tai differentiaalilaskenta. Usein oikeat järjestelmät ovat kuitenkin sen verran monimutkaisia, että pelkästään matemaattisten menetelmien hyödyntäminen on hyvin haastavaa tai lähes mahdotonta. Näissä tapauksissa tarvitaan tietokonepohjaista simulointia järjestelmän toiminnan jäljittämiseen. [1]

Uusia järjestelmiä tai laitteita hankittaessa simulointimalleja käytetään tukemaan yrityksen johdon päätöksiä järjestelmän hankkimisesta. Malleista saadaan kaikki tarvittava tieto siitä, miten järjestelmät suoriutuvat jokapäiväisessä käytössä. Mallintamista ja simulointia voidaan käyttää yhdessä muiden työkalujen kanssa tuotannon suunnittelussa ja parantamisessa. Simuloimalla saadaan esille esimerkiksi valmistuksen pullonkaulat, työvoiman tarve, koneiden käyttöasteen, mahdolliset konerikot ja niiden vaikutuksen, sekä layoutin yleiskuvan. [3]

### 4.1 Määritelmiä

#### 4.1.1 CAD

CAD (Computer-aided Design) eli tietokoneavusteinen suunnittelu on tietokoneohjelmiston käyttöä suunnittelutyössä. Tyypillisimmät CAD-järjestelmien käyttökohteet ovat

erilaiset geometriset mallinnustehtävät ja piirustusten tai dokumenttien tuottaminen. 3D-CAD-suunnittelu on tietokoneavusteista suunnittelua, jossa kappaleesta valmistetaan ohjelmallinen malli, jota voidaan tarkastella eri kuvakulmista. [5]

#### 4.1.2 CAE

CAE (Computer-aided Engineering) on yleisnimitys tietokoneavusteiselle insinööriölle, kuten suunnittelu, valmistus ja tekninen laskenta. Usein esimerkiksi simulointien yhteydessä puhutaan CAE:sta, kun CAD:sta puhuttaessa tarkoitetaan pelkkää tietokoneavusteista piirtämistä. [5]

#### 4.1.3 PLM

PLM (Product Lifecycle Management) eli tuotteen elinkaaren hallinta pyrkii ohjelmistokokonaisuuksien avulla hallitsemaan kaikki tuotteeseen liittyvät tiedot ja suunnitteluprosessit. Pyritään ottamaan huomioon tuotteen elinkaaren kaikki osa-alueet, kuten suunnittelu, valmistus, käyttö, käytöstä poisto ja kierrätys. [10]

#### 4.1.4 Bottom-up-suunnittelu

Valmistetaan ensin kokoonpanon yksittäiset osat ja näistä kootaan kokoonpano. Perinteisesti CAD-järjestelmissä käytetään tätä tapaa. [5]

#### 4.1.5 Top-down-suunnittelu

Yleisemmin tuotekehittäjiöiden käytössä oleva tapa, jossa ensin aloitetaan suuremmasta kappaleesta, joka jaetaan pienempiin osiin niin kauan, että tarvittava yksityiskohmainen tarkkuus saavutetaan. [5]

#### 4.1.6 OPC

OPC (Open Connectivity via Open Standards) eli avointa liitettävyyttä avoimilla standardeilla on ohjelmointirajapinta standardi, jonka avulla Windows-ohjelmat voivat kommunikoida teollisuuden laitteiden avulla. OPC systeemi koostuu serveri-client-parista. OPC-serveri on ohjelma, joka muuntaa esimerkiksi kenttälaitteelta logiikalle tulleen

viestintäprotokollan OPC-protokollaksi. OPC-client-sovellus voi olla mikä tahansa ohjelma, joka on tarkoituksena yhdistää esimerkiksi kenttälaitteeseen. OPC-client käyttää OPC-serveriä saadakseen tai lähettääkseen dataa laitteistolle.

OPC on avoin standardi ja näin valmistajille edullinen käyttää. Laittevalmistajien täytyy tehdä ainoastaan yksi OPC-serveri laittelleen ja tämä pystyy kommunikoimaan minkä tahansa OPC-clientin kanssa. OPC-tekniikan avulla erilaiset ja eri valmistajien tekemät laitteet ja sovellukset voivat kommunikoida keskenään käyttämällä avointa standardisoitua rajapintaa. Teollisuudessa OPC-tekniikkaa käytetään yleisesti esimerkiksi valvomosovellusten ja kenttälaitteita välisessä tiedonsiirrossa.

OPC Foundation on yhteenliittymä, joka luo ja ylläpitää OPC-standardia. OPC Foundationin tehtävä on varmistaa yhteentoimivuus automaatiossa luomalla ja ylläpitämällä avoimia spesifikaatioita, jotka standardisoivat kaikenlaisen datan saamisen eri toimittajien järjestelmien sekä eri tuotannon laitteiden välillä. [11;12]

## 4.2 Reaaliaikainen simulointi

Reaaliaikaisen simuloinnin avulla voidaan simulointi suorittaa lähes välittömästi niissä tapauksissa, joissa tarvitaan nopeita päätöksiä suoritettavista toimista. Näillä nopeilla päätöksillä voi olla suuri vaikutus koko järjestelmän suorituskyykyyn. Esimerkiksi jos ollaan tekemässä erämuotoista tuotantoa, voidaan simulointia käyttää apuna sen määrittämiseksi, missä järjestyksessä eriä kannattaa tehdä. Tai jos jokin avainosa valmistuslaitteista vikaantuu, voidaan tehdä nopeasti suunnitelma vian vaikutuksen minimoimiseksi. [3]

On olemassa kaksi tapaa toteuttaa reaaliaikainen simulointi niin, että mallin rakentamiseen ja tiedonkeruuaikoihin ei mene liikaa aikaa. Nämä ovat simulointimalli, joka on koko ajan synkronoitu tehtaan tuotannon kanssa ja simulointimalli, joka rakennetaan tarvittaessa automaattisesti tehtaan tietokantojen pohjalta. Seuraavassa perehdytään erikseen kumpaankin tapaukseen. [3]

### 4.2.1 Synkronoitu malli

Pysyvä, aina päällä oleva, tehtaan tuotannon kanssa synkronoitu malli pyrkii matkimaan oikean tehtaan käyttäytymistä. Malleja päivitetään ja synkronoidaan jatkuvasti

tehtaalta saatujen tietojen kanssa. Olemassa olevasta mallista voidaan aina tehdä kloonieja tarvittaessa päätöksenteon helpottamiseksi. [3]

Synkronoituun malliin liittyy kuitenkin ongelmia, jotka täytyy ratkaista ennen kuin tätä voidaan alkaa käyttää yleisemmin teollisuuden apuna. Suurin ongelma on yleensä ajallaan olevan tiedon saaminen tehtaalta. Päivitystiheys voi olla liian alhainen, jolloin dataa ei saada kuin esimerkiksi kerran päivässä. Data voi myös olla puutteellista tai liian heikkolaatuista. Tällöin kaikkea simuloinnin tarvitsemaa dataa ei ole saatavilla, eikä sitä voi tuottaa automatisoidusti tehtaan muista saatavilla olevista tiedoista. Ongelmana voi myös olla selvän määritelmän tehtaan tilasta puuttuminen. Täytyy tietää täsmällisesti minkälaista dataa täytyy kerätä, jotta saataisiin mahdollisimman selkeä kuva tehtaan tilasta. [3]

Ennen datan keräämisen aloittamista pitää tietää tarkkaan, minkälaista dataa tehtaalta tarvitaan synkronoidun simuloinnin suorittamiseen. Riippuu tietysti simuloijan ja simuloinnin tavoitteista, onko näillä seikoilla vaikutusta lopputulokseen. Jos tarkoituksen on tehdä tarkka simulaatiopohjainen aikataulutussjärjestelmä, tulee tehdasdatan olla todella tarkkaa, laadukasta ja tarkoituksenmukaista, tai ainakin olla arvio aikataulussa olevasta virheestä, jonka on aiheuttanut väärä, puuttuva tai myöhään tullut data. Toisaalta, jos tarkoituksen on tehdä vain suuntaa-antava yleiskatsaus tehtaan käyttäytymisestä lähipäivinä, ei edellä mainituilla ongelmilla välttämättä ole suurtakaan vaikutusta lopputulokseen. Synkronoituja malleja tehtäessä ongelmana on myös markkinoilla olevien simulointiohjelmistojen vähyys, jotka kykenisivät reaaliaikaiseen toimintaan ja kloonien tekemiseen alkuperäisestä simulointimallista. [3]

#### 4.2.2 Automaattisesti rakennettava malli

Tarvittaessa automaattisesti rakennettavassa mallissa luodaan simulointimalli suoraan tehtaan tietokannan pohjalta. Näin pyritään kiertämään jatkuvasti synkronoidun mallin edellisessä kappaleessa mainittuja mahdollisia ongelmakohtia. Kun malli rakennetaan ainoastaan tehtaan tietokantojen pohjalta, ei itse simulointiohjelmiston tarvitse olla läheskään niin monimutkainen. [3]

Simulointimallit voidaan tehdä alustetuista tekstitiedostoista eli tarvitaan ainoastaan sovellus, joka hakee nykyisen tehtaan datan tietokannoista ja muuttaa sen simulointimalliohjelman ymmärtämään muotoon. Automaattisesti rakennettava simulointimalli

sopii yleensä paremmin ei-aikakriittisiin päätöksiin. Mallin rakentamiseen saattaa mennä huomattavasti aikaa, sillä suuria määriä tehtaan dataa on haettava samanaikaisesti. Jos halutaan nopeuttaa mallin rakentamista, ei dataa kannata säilyttää raakamuodossa, vaan jo valmiiksi esikäsiteltynä ja pakattuna valmiina simuloimiskäyttöön. [3]

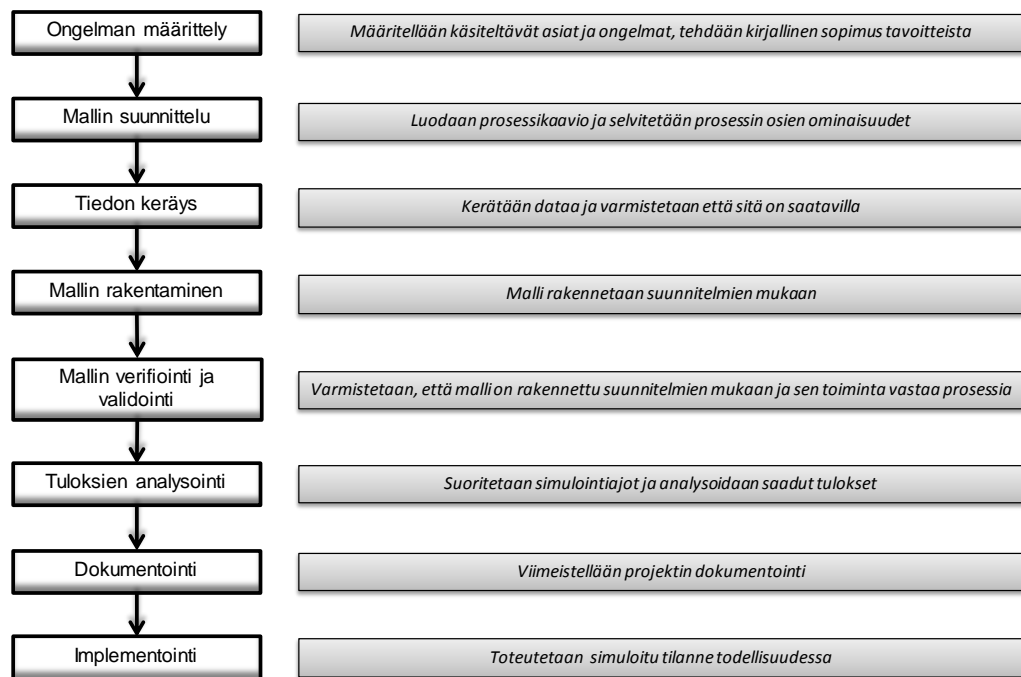
### 4.3 Simulointiprojektin vaiheet

#### 4.3.1 Yleistä

Simulointiprojektissa käydään yleensä projektista riippumatta läpi tietyt vaiheet. Alla olevassa kuvassa 2 esitellään tyypillisen simulointiprojektin vaiheet tyypillisessä järjestyksessä. Usein joudutaan palaamaan edeltäviin vaiheisiin, kun saadaan lisätietoa systeemistä ja sen toiminnasta. Jokainen vaihe on kuitenkin hyvä käydä läpi, sillä ne kaikki ovat tärkeitä onnistuneen lopputuloksen kannalta. [3;13]

Kuvassa 2 on määritelty laajemmin simulointiprojektin kulku jaettuna kahdeksaan eri vaiheeseen. Simulointiprojekti voidaan jakaa myös kolmeen suurempaan kokonaisuuteen: mallin suunnittelu, mallin kehittäminen ja mallin käyttöönotto. Näitä kolmea kokonaisuutta ja niiden tärkeimpiä kohtia kuvaillaan seuraavissa kappaleissa. [3;13]





Kuva 2. Valmistuksenjärjestelmän simuloimiseen kuuluvat vaiheet. [13]

#### 4.3.2 Mallin suunnittelu

Mallin suunnittelun alussa on koko projektin tärkein kohta, jossa määritellään kyseessä olevat asiat ja ongelmat. Ongelman määrittely määrää koko loppuprojektin kulun ja projektin päämäärät. Tässä kohtaa määritellään projektiin osallistujat, projektin päämäärät rajataan ja tehdään lopullinen projektisuunnitelma. Nämä määrittelyt ovat hyvin tärkeitä projektin lopputuloksen ja erityisesti mallin tarkoituksenmukaisuuden kannalta. Määrittelyvaiheessa määritellään muun muassa se, miten yksityiskohtainen mallista halutaan. Tämä on tärkeää sen vuoksi, että ei tehdä liian yksityiskohtaista, mutta liian raskasta, vaikeasti ymmärrettävää ja testattavaa mallia, joka ei vastaa käyttötarkoitusta. Tärkeää on kirjata selkeästi ja täsmällisesti projektin tavoitteet ylös, että vältetään väärinkäsityksiltä ja vääriä olettamuksilta projektin edetessä. On tärkeää, että projektin osallistujat pääsevät keskittymään olennaiseen. [3;4;13]

Seuraavaksi aloitetaan itse mallin suunnittelu luomalla yleensä simulointimallin toimintaa kuvaava prosessikaavio. Se helpottaa simulaation vaiheistuksen suunnittelussa ja yleiskuvan saamisessa järjestelmästä sekä sen osien toiminnasta. [13]

#### 4.3.3 Mallin kehittäminen

Kun simulointiprojektin kulku on tarkkaan ja täsmällisesti määritelty ja dokumentoitu, voidaan siirtyä itse mallin kehittämiseen. Se aloitetaan datan keräämisellä ja varmistamisella, että simuloinnin suorittamiseen tarvittavaa dataa on saatavilla. On myös tärkeää, että data ei ole vanhentunutta ja että se on oikeassa muodossa. Datan kerääminen saattaa osoittautua haasteelliseksi ja aikaa vieväksi vaiheeksi, varsinkin jos asiakas ei ole täysin tietoinen saatavilla olevasta datasta ja sen laadusta. [4;13]

Mallin rakentaminen aloitetaan mallinnustavan ja käytettävän mallinnusohjelman valinnalla. Vaikka mallinnustavan valinnalla ei voida saavuttaa suurta tehokkuushyötyä tätä kyseistä mallia rakennettaessa, voidaan mallinnustavan valinnalla tehdä suuri ero seuraavissa mallinuksissa ja mallin suoritusajoissa. Vaikka nykyaikaiset mallinnusohjelmat ovat hyvinkin tehokkaita, ei mallin rakentamiseen varata usein tarpeeksi aikaa, vaikka tiedetään sen vievän yleensä paljon aikaa projektissa. Mallin rakentaminen aloitetaan yleensä hyvin yksinkertaistetusta versiosta, jonka jälkeen mallia tarkennetaan osa kerrallaan halutulle tarkkuuden tasolle. [3;13]

Seuraavaksi simulointimalli verifioidaan ja validoidaan. Verifiointissa testataan, että malli on tehty oikein ja että sen on valmis. Validoinnissa taas varmistetaan, että tehty malli on täsmällinen kuvaus kyseessä olevasta oikeasta järjestelmästä. Näillä toimilla varmistetaan, että malli toimii niin kuin on tarkoitus, ja että se täyttää muutenkin suunnitteluvaiheessa määritellyt tarkkuus- ja vastaavuusmäärittelyt. [13]

#### 4.3.4 Mallin käyttöönotto

Mallin käyttöönotossa testataan ensin simulointimallia. Näistä testeistä saatuja tuloksia analysoidaan, dokumentoidaan ja lopuksi niitä sovelletaan päätöksentekoon. Simulointimallin testauksessa pyritään tehostamaan mahdollisuuksien mukaan mallin suorittamista, sillä monimutkaisten valmistusjärjestelmien simuloiminen voi olla hidasta puuhaa. Simulaatioaikojen hitaudesta johtuen ongelmanratkaisusyklien määrää voi olla rajoitettu varsinkin monimutkaisemmissa järjestelmissä. Hyvin usein käyttöönottotestin jälkeen joudutaan tekemään muutoksia mallin toimintaan. Tämä muutoksien tekeminen voi jatkua joskus lähes loputtomiin ja mallin hiomiseen voidaan käyttää huomattavasti aikaa. Tässä vaiheessa on tiedettävä, milloin kannattaa lopettaa mallin kehittäminen. Projektin määrittelyvaiheen tärkeys korostuu tässäkin vaiheessa projektia. [13]

Tuloksen analysoinnissa on tärkeää ymmärtää järjestelmän toiminta ja esittää saadut tulokset ymmärrettävällä tavalla. Useimmiten, jos halutaan vertailla useimpien muuttujien vaikutusta toisiinsa, joudutaan tuloksen esittämään graafisessa muodossa niiden lukemisen ja ymmärtämisen helpottamiseksi. Jos järjestelmältä tulevaa dataa ei ymmärrä, ei sitä pysty analysoimaan. [13]

Jos projektista ja mallista on tehty hyvä dokumentointi, se tarjoaa tarvittavat tiedot mallin jatkuvaa käyttöä varten, sekä tukee tarvittaessa mallin lisärakentamista ja analysointia. Projektin edistymisraportit ovat tärkeää dokumenttia projektin etenemisestä. Näistä raporteista selviää mitä kulloinkin tehtiin, mitkä toimet toimivat ja mitkä eivät. Edistymisraportit voivat olla hyvä keino pitää projekti aikataulussa. Erityisesti alkuvaiheessa säännöllinen raportointi projektin etenemisestä ja suunnitelmista on tärkeää. Jos ongelmat ja projektiin liittyvät väärinkäsitykset huomataan jo alkuvaiheessa, niiden vaikutus saadaan minimoitua ja niiden korjaaminen on huomattavasti helpompaa ja halvempaa kuin projektin edettyä pidemmälle. [4;13]

Implementoinnissa eli toimeenpanossa toteutetaan simuloitu tilanne todellisuudessa. Loppujen lopuksi projektin onnistuminen määritellään sen pohjalta, kun asiakas tekee jonkin konkreettisen toimenpiteen simuloinnin pohjalta. Projekti on onnistunut, jos joko järjestelmän suorituskyky paranee tai saadaan tehtyä uusi alussa määritellyllä tavalla toimiva järjestelmä. [4]

Liian usein projekteissa simulointia käytetään apuna vasta silloin, kun tajutaan, että saatetaan olla jo ongelmissa. Useimmiten tämä tapahtuu vasta juuri ennen kuin ollaan lyömässä lukkoon järjestelmän malli ja tilaamassa itse järjestelmä. Yleensä tällöin ei ole kunnolla aikaa mallin rakentamiseen, eikä mallia saada välttämättä ajoissa valmiiksi. Vaikka simuloinnista saataisiin vihjeitä, että järjestelmässä voi olla ongelmia, ei tätä tietoa ehditä käyttää hyväksi ja näin koko suoritettu simulointi on ollut käytännössä turhaa. [4]

#### 4.3.5 Ongelmia

Ongelmana simulointimallien käyttämisessä laajemmin teollisuudessa on se, että kestää liian kauan suunnitella, rakentaa ja toteuttaa simulointimalleja, sekä kerätä ja analysoida dataa. Näin niitä ei pystytä käyttämään valmistuksen päätöksenteon apuna. Usein simulointimalleja käytetään vain yksittäisten hankkeiden päätöksenteon apuna ja

uuteen projektiin joudutaan rakentamaan malli aina tyhjästä. Tämä vaatii aina huomattavan määrän työtä. Esimerkiksi jos simulointimalleja käytetään laitehankintojen suunnitteluun tai arvioimaan suunniteltujen muutosten vaikutusta materiaalivirtojen hallinnassa, ei malleja välttämättä käytetä enää koskaan sen jälkeen kun tarvittavat päätökset on tehty. Toisaalta useissa simulointiohjelmissa on joko valmiina, tai erikseen ladattavissa erinäisiä kirjastoja ja valmiita malleja toimilaitteista, kuten kuljettimista ja roboteista. [3;13]

Kahden eri henkilön samasta järjestelmästä tekemä mallit eroavat usein toisistaan jonkin verran, vaikka niissä olisi paljonkin samaa. Simulointimallien rakentaminen on oma taiteenlajinsa ja se vaatii rutkasti käytännön kokemusta. Simulointi ja tulosten analysointi voi joissakin tapauksissa olla hyvin aikaa vievää ja kallista. Jos näissä tapauksissa pyritään säästämään väärissä kohdissa, voivat tulokset olla vääriä tai soveltumattomia. Myös simulaatiota kohtaan voidaan ladata helposti liian suuret ja kohtuuttomat odotukset. Jos järjestelmä on liian monimutkainen, kuten ihmisten käyttäytyminen jossakin tapauksissa, ei simulointi välttämättä ole oikea lähestymistapa. [3;13]

#### 4.3.6 Hyötyjä

Simuloinnin avulla pystytään simuloimaan tuotantojärjestelmän toiminta usean vuoden ajalta hetkessä, tai tarvittaessa voidaan aikaa hidastaa niin, että saadaan tarkka kuva jostain järjestelmän tietystä osasta tai toiminnasta. Simulointiympäristössä kaikkea pystytään seuraamaan ja valvomaan tarkasti, näin myös ongelmien ratkominen erityisesti hyvin monimutkaisissa järjestelmissä helpottuu. Pystytään ymmärtämään paremmin eri osien vuorovaikutusta ja näiden vaikutusta koko järjestelmän toimintaan. Pystytään testaamaan ja tutkimaan järjestelmän toimintaa ennen kuin itse järjestelmään on sijoitettu yhtään pääomaa. Näin välttyään tai ainakin vähennetään kalliita muutoksia tai korjauksia, joita jouduttaisiin tekemään jo asennettuun järjestelmään. Esimerkiksi järjestelmässä esiintyvät pullonkaulat voidaan huomata jo etukäteen. Pystytään myös tutkimaan teoriassa tai käytännössä mahdollisesti vaarallisia järjestelmiä ilman taloudellisia tai fyysisiä riskejä. [1;13]

Kun järjestelmä on jo rakennettu, voidaan sitä käyttää tutkimaan eri toiminatapoja ja mahdollisten parannuksien tekemistä järjestelmään ilman että oikeaa järjestelmää häiritään. Valmis malli on myös hyvä tapa kouluttaa työntekijöitä. Säästetään rahaa, kun työntekijät tekevät virheitä simulaatiomallin kanssa ja oppivat näistä virheistä. Työnteki-

jät oppivat myös järjestelmän yleistöinnasta ja käytöstä. 3D-malli voi auttaa hahmotamaan koko järjestelmää ja sen toimintaa, tämä voi olla tehokkaampi ja helpompi keino kuin selittää sen toimintaa muilla tavoin. [1;13]

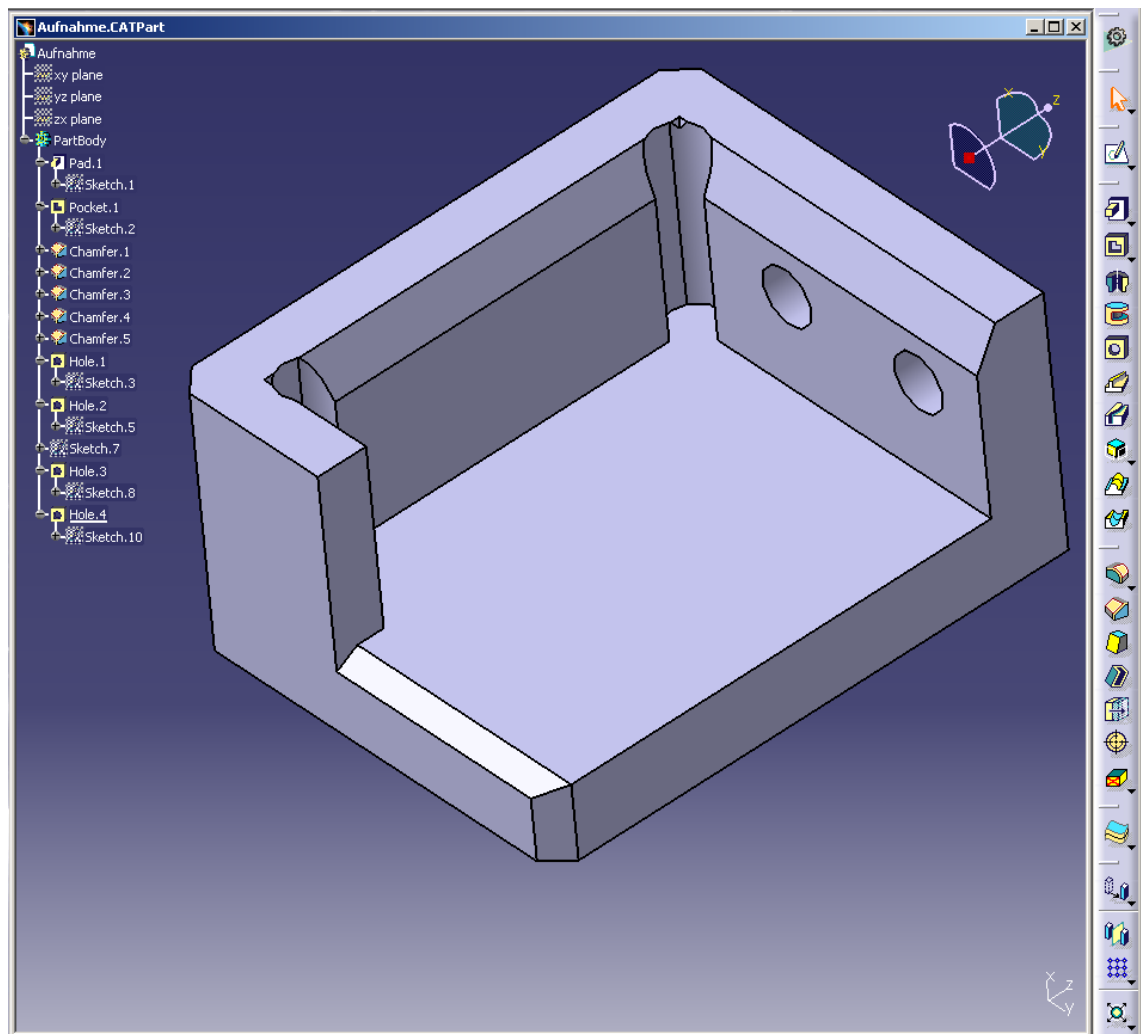
## 5 Toteutus

### 5.1 Osien luominen

Mallin osat tehtiin yksinkertaisesti Part Design -työpöydällä CATPart-tiedostoiksi. Mallien mitat ja muodot saatiin linjaston toimittajan tarjoamista saksankielisistä piirustuksista. Osien luominen aloitetaan avaamalla uusi tyhjä CATPart-tiedosto. Part Design -työpöytä avautuu samalla, ja käytettävissä olevat työkalut näkyvät oletusarvoisesti ohjelman oikeassa reunassa kuvan 3 mukaisesti.

Osien tekeminen aloitetaan aina valitsemalla joko jokin kolmesta xy-, yz- tai zx-tasoista tai tekemällä itse tason haluttuun paikkaan Wireframe and Surface Design -työpöydältä löytyvällä Plane-työkalulla. Valitulle tasolle piirretään halutun muotoinen suljettu sketchi. Constrain-työkalulla voidaan määrittää kappaleella tarkat mitat. Kun sketchi on täysin mitoitettu, se muuttuu vihreäksi. Tehty profiili valitaan ja pursotetaan haluttuun suuntaan. Muodostuvaa 3D-kappaletta voidaan muokata edelleen piirtämällä sketchejä joko edellä mainituille tasoille tai valitsemalla suoraan jokin muodostuneen kappaleen reuna.

Ohjelmassa on myös useita aikaa ja hermoja säästäviä pursotetun kappaleen muokkaamiseen tarkoitettuja työkaluja, joihin ei tarvitse tehdä erillisiä sketchejä. Esimerkiksi jos kappaleessa on paljon toistuvia rakenteita, voidaan mirror- ja pattern-työkaluilla tehdä vain puolikas tai osa kappaleesta ja muodostaa tästä suurempi kappale vähällä vaivalla. Chamfer- ja erilaisilla fillet-työkaluilla saadaan helposti kappaleiden reunoihin halutunlaiset viisteet ja pyöristykset.



Kuva 3. Osa kääntölaitteesta Part Design –työpöydällä.

## 5.2 Kokoonpano

Assembly Design -työpöydällä voidaan useammasta tehdystä CATPart-tiedostoista koota yksi CATProduct-tiedosto. Kokoaminen aloitetaan avaamalla tyhjä CATProduct-tiedosto ja lisäämällä siihen tehdyt osat insert existing component -toiminnolla. Osat sijoittuvat lisättäessä siihen kohtaan, mihin ne on aikaisemmin luotu xy-, yz- tai zx-tasojen suhteen. Osia pystytään siirtämään haluttuun paikkaan muun muassa siirtämällä oikeassa yläkulmassa oleva kompassi halutun osan päälle. Kompassi muuttuu vihreäksi ja osaa voidaan pyöritellä ja liikutella haluttuihin suuntiin. Siirrettäessä jokaisen osan tasot seuraavat osiaan.

Kokoonpanojen luomiseen on olemassa erilaisia tapoja ja työkaluja. Tässä tapauksessa ehkä helpoin vaihtoehto on käyttää snap-työkalua. Valitaan ensin liikutettavasta

osasta jokin sivu, reikä tai muu piirre, ja vastaavasti valitaan toisesta osasta jokin piirre, jolloin liikutettava osa liikkuu tämän mukaisesti. Kun snap-toimintoja toistetaan tarpeeksi, saadaan osat haluttuihin kohtiin. Osien yhteenliittämiseen voidaan käyttää myös erilaisia constraints-työkaluja. Nämä työkalut luovat erilaisia rajoituksia ja relaatioita osien välille. Constraints-työkaluja voidaan käyttää myös osoittamaan mekaanisia suhteita eri osien välille, jolloin niiden liikkeet rajoittuvat. Käytin kokoonpanoja tehdessä pelkästään snap-työkalua, koska ei haluttu rajoittaa osien liikettä toisiin osiin nähden, kunnes vasta myöhemmin mekanismeja kokoonpanoon tehdessä.

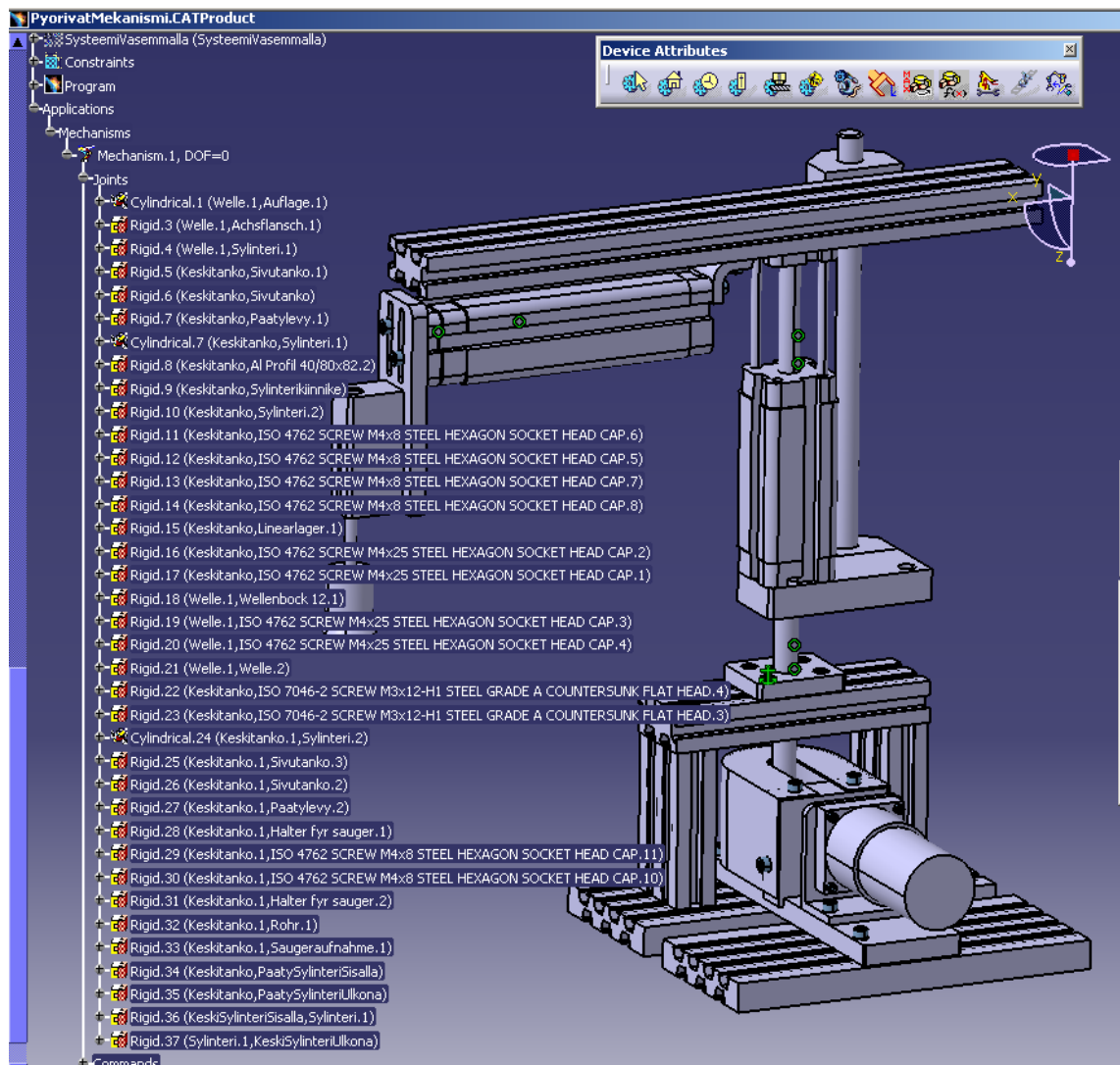
### 5.3 Mekanismit

Mekanismien tekoon tarvittavat työkalut löytyvät Device Building -työpöydältä. Device Building -työpöytä sisältää valikoiman työkaluja, joilla voidaan mallintaa valmistusprosesseissa yleisimmin käytettäviä mekaanisia järjestelmiä, kuten esimerkiksi robotin päätelaitteet, paikannuslaitteet, jyrsimet, sorvit ja teollisuuden robotit.

Device Building -työkaluilla pystytään luomaan mekanismeja, joihin sisältyy niveliä (joints). Nivelet voivat sisältää erilaisia mekaanisia ja geometrisiä rajoituksia (constraints), esimerkiksi kiinteän akselin ympäripyöriminen ja osan pinnan seuraaminen. Device Attributes -työkaluilla voidaan mekanismeihin määritellä kotipositioita (home positions), eli paikkoja, mihin laitteen halutaan liikkuvan. Mekanismeihin pystyy määrittelemään liikkeen ominaisuuksia, kuten liikkumisrajoja, -aikoja ja kiihtyvyyden rajoja.

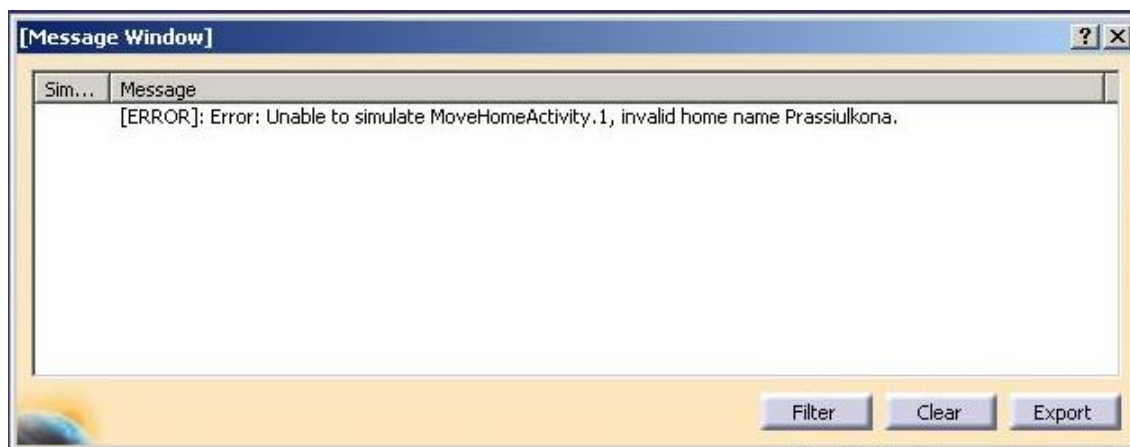
Mekanismien tekeminen toteutetaan tekemällä ensin uusi mekanismi (new mechanism) ja lisäämällä siihen kiinteä osa (fixed part). Kun jokin osa on määritelty liikkumattomaksi, voidaan mekanismiin lisätä niveliä. Device Building -työpöydällä voidaan määritellä ja muokata 11 erilaista niveltyyppiä, tässä työssä käytettiin lähinnä cylindrical-, rigid- ja prismatic-niveliä. Työssä tehtiin yksi mekanismi toimilaitetta kohden aina kun se oli mahdollista. Esimerkiksi kuva 4: Yhden mekanismin alle on kasattu useampia niveliä. Ensimmäinen tehtiin yksi cylindrical nivel, johon liitettiin rigid aina tarvittavat rigid nivelet, kunnes taas tuli vastaan seuraava pyörivä cylindrical nivel. Näin jatkettiin niin kauan, kunnes päästiin toimilaitteen loppuun. Kyseinen kolmen vapausasteen toimilaitte koostuu 36 nivelestä.





Kuva 4. Niveliiä mekanismissa.

Yritettäessä tehdä linjaston liukuhihnan kahta pyörivää mekanismia yhteen CATProduct-tiedostoon, törmättiin kuvan 5 mukaiseen virheilmoitukseen. Ohjelma ei pystynyt liikuttamaan mekanismeja kotipositioihin (home position), kun ainoastaan silloin kun tiedostossa oli vain yksi mekanismi. Kyseisestä ongelmasta selvittiin lisäämällä CATPart-tiedoston sisälle kaksi erillistä CATPart-tiedostoa, joihin tiedoston osat jaettiin niin ja kumpaankin tehtiin oma mekanisminsa.



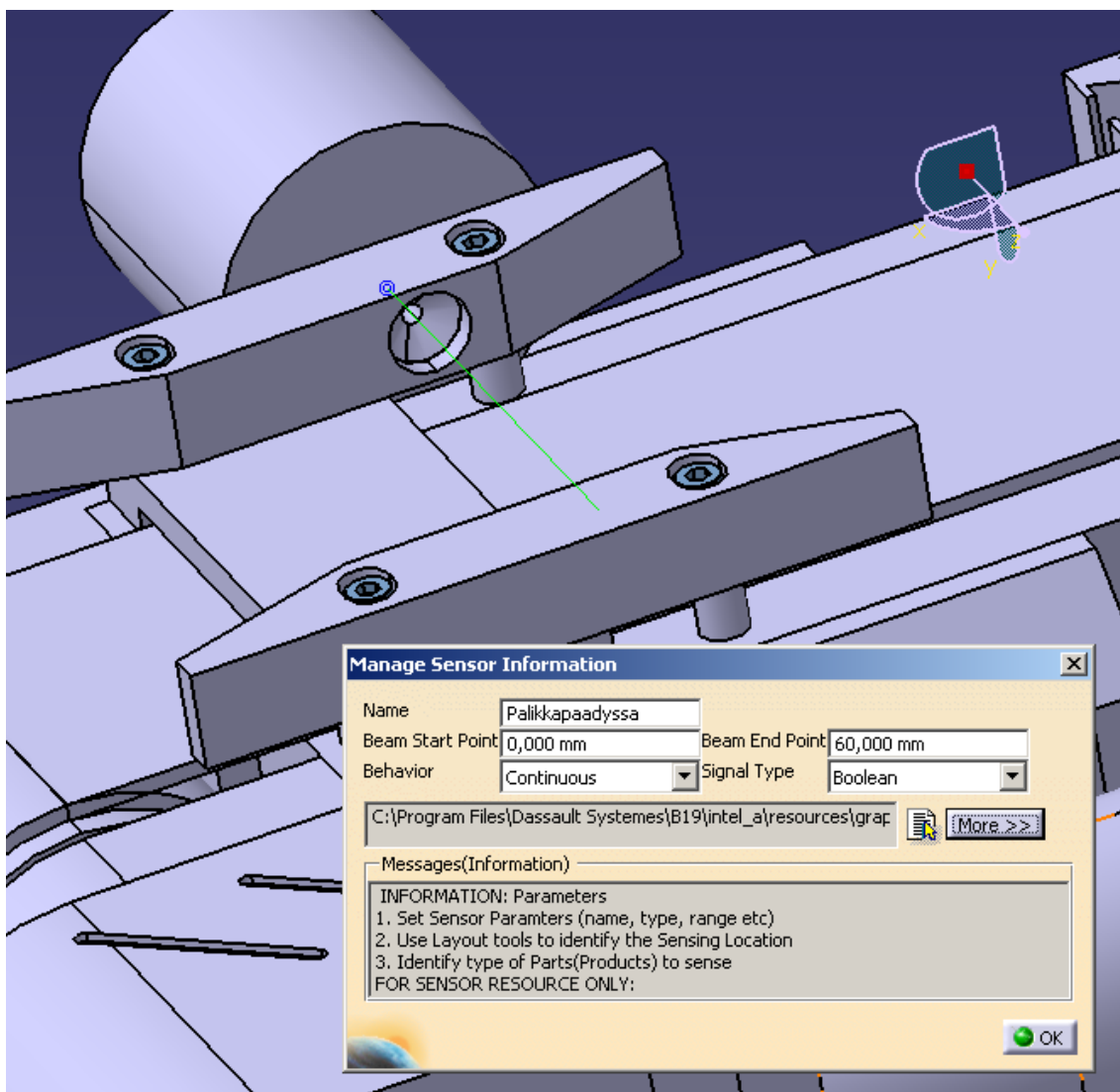
Kuva 5. Virheviesti mekanismeja luodessa.

#### 5.4 Anturit

CLM Device Logic Design -työpöydällä voidaan määritellä ja suunnitella laitteiden käyttäytymistä. Tätä kutsutaan yleisesti laitteen sisäiseksi logiikaksi. Sisäinen logiikka koostuu jonkin laitteen porteista, eli tuloista ja lähdöistä, sekä päälohkosta, joka sisältää ohjelman laitteen kinematiikan ohjaamiselle. Laite, jossa on sisäinen logiikka, tunnustetaan älylaitteeksi (smart device). Yksittäisistä älylaitteista voidaan koota monimutkaisempia järjestelmiä, kuten linjastoja ja valmistusasemia. Sisäinen logiikka voidaan linkittää virtuaaliseen tai oikeaan logiikkaan.

CLM Device Logic Design -työpöydältä löytyvällä create a resource sensor -työkalulla pystytään malleihin lisäämään antureita. Antureille voidaan määrittää tunnistettavat kappaleet ja matka sekä valita signaalin tyyppi boolean vai integer. Boolean anturit palauttavat true, jos osa havaitaan ja integer anturit palauttavat tiettyyn osaan määritellyn integer arvon. Antureita voidaan käyttää datan keräämiseen simulointia suoritettaessa ja niitä voidaan käyttää oikeiden antureiden tavoin tunnistamaan kappaleita ja toimilaitteiden asentoa.

Tässä työssä käytettiin pelkästään kuvan 6 mukaisia boolean signaalia palauttavia antureita. Antureille määriteltiin, mitä osia niiden on tarkoitus tunnistaa. Joissakin tapauksissa voitiin valita kaikki osat tunnistettaviksi.



Kuva 6. Anturin määrittelyä kuljetushihnan lopussa.

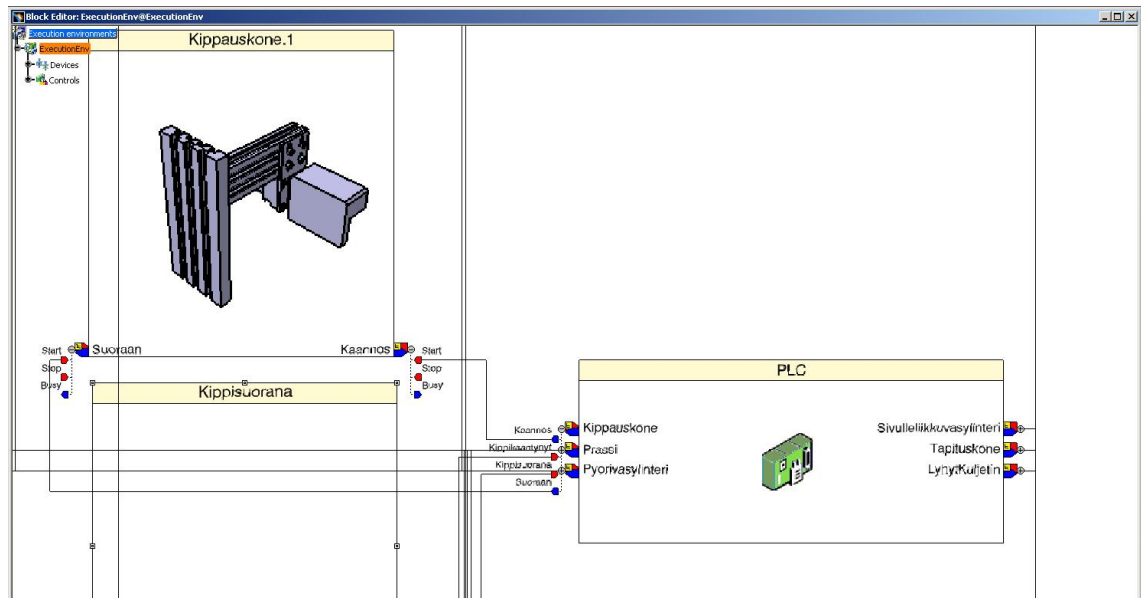
## 5.5 OPC

CSM Device Control Connection -työpöydällä voidaan tehtyjä 3D-malleja testata yhdistämällä ne käytössä oleviin tuotantojärjestelmiin. Logiikkaohjelmistoa voidaan testata ja varmentaa 3D-mallin visualisoinnin avulla. Ohjelmassa on valittavissa tiedonsiirtoprotokolliksi Modbus/TCP, OPC tai S7-PLCSIM. Tässä työssä käytettiin systeemin testaamiseen OPC-liityntää. OPC-serverinä työssä käytettiin ei-tuotantokäyttöön ilmaista MatrikonOPC Server for Simulation-ohjelmaa. Vaikka kyseinen ohjelma on ainoastaan OPC server-simulaattori, eikä se sisällä esimerkiksi logiikan simulointi mahdollisuutta, on se tässä tapauksessa täysin riittävä yhteyden testaamiseen.

Jokainen toimilaite tehtiin ensin älylaitteeksi lisäämällä niihin sisäinen logiikka. CLM Device Logic Design -työpöydältä valittiin generate internal logic automatically. Automaattinen sisäisen logiikan luonti onnistuu tällä työkalulla yksinkertaisesti, kun aina kunkin laitteen tehtävät (device tasks) on luotu kotipositioiden (home positions) perusteella. Automaattinen sisäisen logiikan luominen tekee ulkoiseen liittämiseen tarvittavat portit automaattisesti. Nämä portit vastaavat simuloitaessa fyysisen järjestelmän portteja.

OPC-yhteyden luominen aloitettiin lisäämällä koko linjaston malliin suoritusympäristö (Execution Environment). Kyseisellä työkalulla monen laitteen tulot ja lähdöt saadaan lisättyä yhteen paikkaan. Näin kaikkia laitteita pystytään tarkkailemaan ja kontrolloimaan samanaikaisesti. Execution Environment määrittelyikkunaan devices kohtaan lisätään jo olemassa olevat sisäiset logiikat. Controls kohtaan lisätään OPC-yhteydellä oleva PLC. PLC properties ikkunassa yhdistetään OPC serveriin ja OPC-tagit saadaan lisättyä external control's IOs listalle.

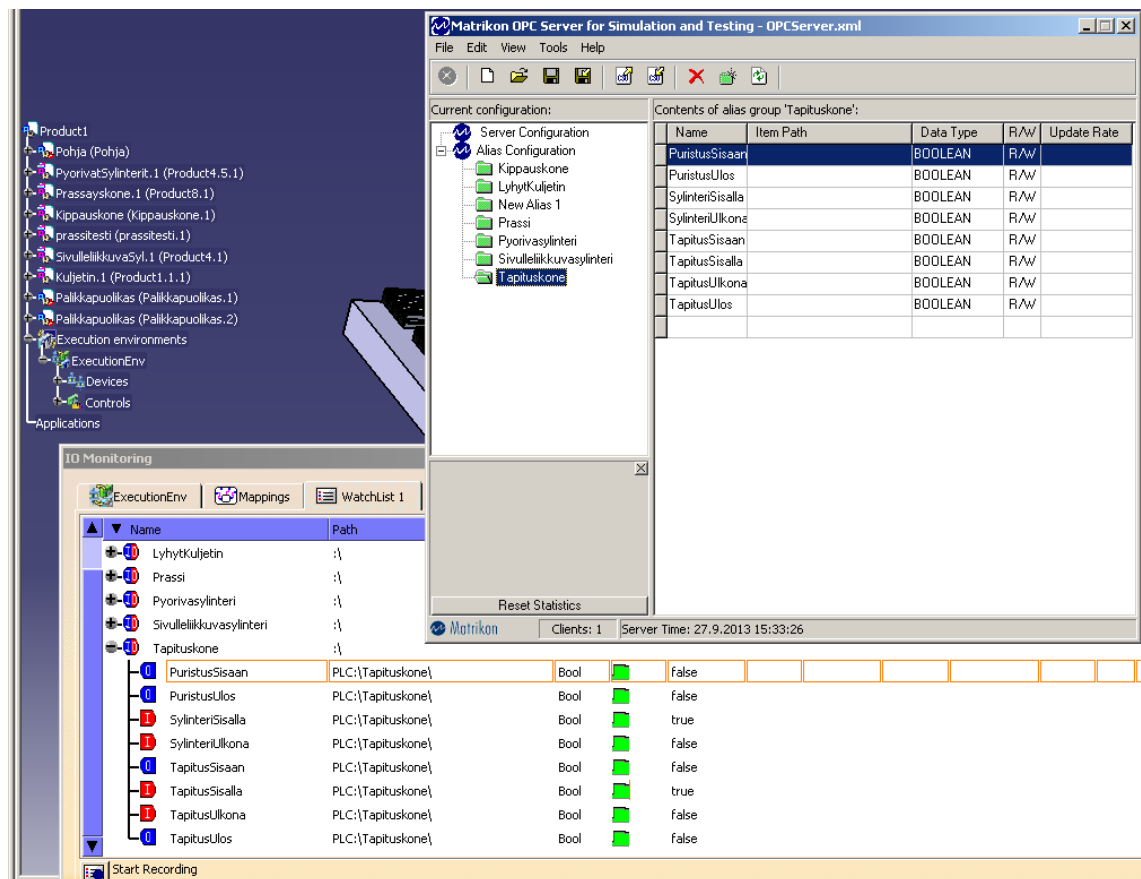
Nyt execution environments kohdan alla on erillisenä devices ja controls, näiden kahden linkittämiseen käytetään open connection editor -työkalua kuvan 7 mukaisesti. Toisin sanoen sisäiset logiikat yhdistetään OPC:hen. Jokainen PLC:ltä tuleva OPC-tagi liitetään vastaavaan toimilaitteen liikkumiseen, tai anturiin manuaalisesti add connection -työkalulla. Lopputuloksena on varsin sekava, mutta toimiva laitteiden ja ohjauksien yhdistäminen.



Kuva 7. OPC-tagien sekavahko linkittäminen antureihin ja laitteiden liikuttamiseen.

## 5.6 Simulointi

Systeemiä, jossa on OPC-yhteys, simuloidaan CSM Device Control Connection -työpöydällä olevalla Execute-työkalulla. Simuloinnin ollessa run-tilassa voidaan IO monitoring -ikkunassa seurata antureiden sekä toimilaitteiden tilaa sekä pakottaa haluttaessa toimilaitteet liikkumaan kuvan 8 mukaisesti.



Kuva 8. MatrikonOPC Server for Simulation ja IO monitoring näkymä simuloinnin ollessa käynnissä.

## 6 Yhteenveto

Insinööriyössä toteutettiin toimiva simulaatiomalli yhdestä kolmasosasta Metropolian automaatiotekniikan laboratoriosta löytyvästä kappaletavaralinjastosta ja sen OPC-rajapinta, jota testattiin OPC-serverin simulointiohjelmalla. Työssä saatiin myös aikaiseksi kaksi eri laboratoriotyötä koulun käyttöön. Toinen töistä liittyi Delmian OPC-rajapinnan luomiseen ja toisessa tutustutaan Delmialla tuotantoprosessin luomiseen.

Kokonaisprojektia ei ikävä kyllä päästy testaamaan johtuen lieivistä ongelmista kahden eri koneen WinCC ja Delmia välisen OPC-yhteyden kanssa. Kaiken kaikkiaan olen kohtuullisen tyytyväinen omaan osuuteeni projektissa ja pääsin asetettuihin tavoitteisiin sekä mallin, että laboratoriotöiden suhteen.

### 6.1 Aikaansaatu kokonaisuus ja vertailu tavoitteisiin

Työn tarkoituksena oli tehdä simulointimalli jo olemassa olevasta kappaletavaralinjastosta. Tätä mallia oli tarkoituksena voida myöhemmin käyttää opetuskäytössä ja laboratorioharjoituksissa. Päädyttiin lopulta muuttamaan määrittelyä sen verran, että tarkoituksen oli tehdä vain kolmasosa linjastosta. Tähän päädyttiin siitä syystä, että itse 3D-mallin piirtäminen oli hyvin aikaa vievää ja loppujen lopuksi hyvin yksitoikkoista. Pyrittiin tekemään mallin osat ja niiden mitat kohtuullisen saatavilla olleiden piirustusten mukaisesti, että koko mallia kasatessa ei törmättäisi uusista mitoista johtuviin ongelmiin.

Koska linjastosta tehtiin vain kolmasosa, ei koko tuotantolinjastoa pystytty simuloimaan, mutta käytössä ollut tietokone oli jo kolmasosa linjastoa käytettäessä niin hidas, että mallin ajaminen oli hyvin tuskallista. Jos olisi haluttu tehdä koko linjaston malli, olisi se ollut lähinnä vain kopioimista jo tehdyistä toimilaitteista, sillä siinä toistuivat hyvin pitkälti samat kuljettimet, sylinterit, anturit ja tarttujat. Toki linjaston tehdyssäkin osassa olisi vielä hiomista, jos siitä haluttaisiin mahdollisimman realistinen ja visuaalisesti näyttävä, mutta nykyinenkin malli riittää kyllä helposti tähän työhön.

Haastavin osuus projektissa oli selvästi mekanismien ja niiden nivelien tekeminen toimilaitteille. Näitä tehdessä menttiin hyvin yritys-erehdys-periaatteella, ja osa toimilaitteista jouduttiin tekemään useampaan kertaan, että saatiin ne toimimaan halutulla tavalla ja ilman virheilmoituksia. Lopputuloksena saatiin kuitenkin loogisesti ja oikealla tavalla toimivat toimilaitteet, joita pystytään ohjaamaan aivan niin kuin oikeita fyysisiä

toimilaitteitakin. Myös simulointimallin anturit saatiin toimimaan halutulla tavalla ja myös ne saatiin testattua OPC-serveri simulaattorin avulla.

Kokonaisprojektia, eli koko tuotantolinjastoa ajatellen, emme päässeet aivan tavoitteisiin Delmialla tehdyn simulointimallin ohjaamisessa WinCC-ohjelmistolla johtuen ongelmista OPC-yhteydessä kahden tietokoneen välillä. Samalla koneella nämä kaksi ohjelmaa toimivat hyvin yhteen, mutta eri koneilla yhteyden toimintaan saaminen olisi vaatinut lisää työtä ja aikaa.

Kaiken kaikkiaan työ oli hyvin mielenkiintoinen ja sopivan haastava, vaikkakin 3D-mallin luominen osa kerrallaan oli hieman turhauttavaa yksitoikkoisuudesta johtuen. Työn aikana opittiin paljon yleisesti simulaatioiden käytöstä kappaletavaraateollisuudessa ja Delmian tarjoamista monipuolisista ominaisuuksista näitä tehdessä. Delmia-ohjelmiston käyttö oli mielestäni pidemmän päälle loogista ja se tarjoaa mahdollisuudet hyvinkin monimutkaisten tuotantojärjestelmien tarkkaan simuloimiseen.

## 6.2 Kehitysmahdollisuudet

Jatkokehityksenä voitaisiin mallintaa puuttuvat kaksi kolmasosaa kappaletavaralinjastosta ja näin pystyttäisiin simuloimaan kappaleiden tuotantoa kokonaisvaltaisesti. OPC-yhteys Delmian ja WinCC:n välille voitaisiin myös toteuttaa niin, että pystyttäisiin toteuttamaan kokonainen virtuaalinen tuotantojärjestelmä. Tuotannosta voitaisiin kerätä historiatietoja tietokantaan ja analysoida näitä OEE-laskennan avulla. Tuotannon simulaatioon voitaisiin tehdä pysäytyksiä ja muita virheitä aivan niin kuin oikeissakin tuotantolaitoksissa tapahtuu. Ennen näitä jatkokehityksiä olisi mielekästä kuitenkin päivittää tietokone tehokkaammaksi, jotta Delmialla tehty simulaatio pyörii sujuvammin ja mallin laajentaminen on kannattavaa, tai ylipäätään edes mahdollista.

## 6.3 Laboratoriotyöt

Työssä valmistui kaksi erillistä laboratoriotyötä koulun käyttöön liittyen tehtyyn simulointimalliin. Nämä olivat nimeltään OPC Delmia, sekä Delmia Process. OPC Delmia -työssä tutustutaan linjastoon kuuluvan kippauskoneen liittämiseen Matrikon OPC Server simulaattoriin. Tarkoituksena on saada aikaan OPC-yhteys ja päästä simuloimaan kahta anturia sekä kipin liikettä edestakaisin.



Delmia Process -laboratoriotyössä on tehty valmiiksi CATProcess-tiedosto, jossa linjaston osaan on tehty yksi prosessilistaus, joka käy läpi koko linjaston osan toiminnan. Tässä laboratoriotyössä ei ole minkäänlaista ulkoista ohjausta eikä OPC-liitäntöjä, vaan ainoastaan prosessilistaus, jota ohjelma käy läpi tuotteiden valmistuksessa. Tämä koko Delmia Process -laboratoriotyö oli irtonainen osa työtä, eikä sitä alun perin ollut tarkoitus tehdä. Tarkoituksena oli vain työn lopuksi saada testattua koko linjastonosan toimintaa yhteen putkeen. Prosessilistausta tehdessä löydettiin kuitenkin Grab- ja Release-komennot, jotka Heikki Hietala insinöörityössään epäili puuttuvan Delmian lisenssistä kokonaan pohjautuen siihen, että niitä ei löytynyt online help-toiminnon osoittamasta paikasta. Nämä toiminnot olivat kuitenkin täysin eri paikassa, ne löytyivät Human Task Simulation -työpöydältä, vaikka ohjelman help väitti niiden löytyvän Device Buildin -työpöydältä.

Näitä Grab- ja Release-komentoja pystyttiin käyttämään linjastoa simuloitaessa tarttumaan poimimaan ja irrottamaan tuotettavia palikoita. Näin ohjelmalla pystyttäisiin simuloimaan jatkuvatoimista tuotantoa yksinkertaisemmin, kun kappaleisiin pystytään tarttumaan ja niistä päästämään irti yhdellä komennolla.

## Lähteet

- 1 Banks, Jerry. Discrete-Event System Simulation. Chapter 1.  
[http://cs.tju.edu.cn/faculty/zenghua/sim\\_ref/chapters\\_of\\_Bank%27s%20book/ch01.pdf](http://cs.tju.edu.cn/faculty/zenghua/sim_ref/chapters_of_Bank%27s%20book/ch01.pdf).
- 2 Delfoi kotisivu. <http://www.delfoi.com/>.
- 3 W. Fowler, John & Rose, Oliver. Abstrakti. Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems.  
<http://www.thesimguy.com/GC/papers/scsgc-02.pdf>.
- 4 Banks, Jerry & Gibson, Randall R. Simulating in the Real World. IIE Solutions 2001. <http://www.diamond-head-associates.com/Websites/enterotech/Images/Simulating%20in%20the%20Real%20World.pdf>.
- 5 Laakko, Timo, ym. 1998. Tuotteen 3D-CAD-suunnittelu. 1. Painos. WSOY Porvoo.
- 6 Delmia online help documentation V5R19. (C:\Program Files\Dassault Systemes\B19doc\English\online\DELMIA\_default.htm).
- 7 CadWorks kotisivu, Dassault Systèmes. <http://www.cadworks.fi/dassault>.
- 8 Delmia. <http://en.wikipedia.org/wiki/DELMIA>.
- 9 Dassault Systèmes kotisivu. <http://www.3ds.com/>.
- 10 Tuotteen elinkaaren hallinta.  
[http://fi.wikipedia.org/wiki/Tuotteen\\_elinkaaren\\_hallinta](http://fi.wikipedia.org/wiki/Tuotteen_elinkaaren_hallinta).
- 11 OPC Foundation. <http://www.opcfoundation.org/>.
- 12 OPC DataHub. <http://www.opcdatahub.com/>
- 13 Banks, Jerry. 1998. Hadbook of Simulation. A Wiley-Interscience publication. USA.

## **Laboratorioharjoitus**

OPC Delmia

Metropolia Automaatiotekniikka

## 1. Johdanto

Tässä työssä on tarkoituksena tutustua Delmia V5R19 ohjelmaan ja sen liittämiseen Matrikon OPC Server simulaattoriin. Delmia on Catia V5 -käyttöliittymän päälle rakennettu digitaaliseen valmistukseen ja prosessisimulointiin erikoistunut ohjelmistokokonaisuus.

Työssä on valmiiksi tehty malli automaatiolaboratoriosta löytyvästä ”kippauskoneesta”. Malliin kuuluu kaksi anturia, sekä toiminnot kipin kippaamiseen ja takaisin suoraan saamiseen. Tarkoituksena on saada nämä kaksi anturia ja kippaustoiminnot liitettyä OPC simulaattoriin.

## 2. Tehtävä

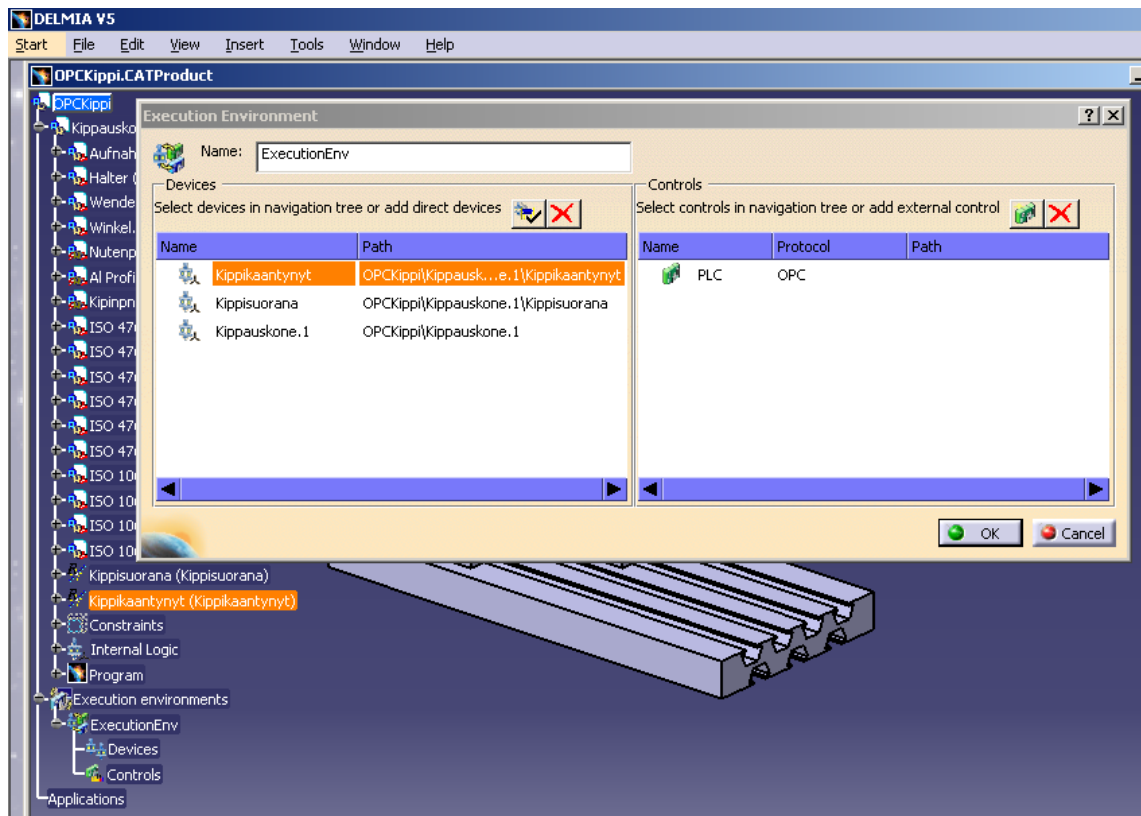
### 2.1 Delmia valmistelua

Käynnistä Delmia V5R19 ohjelmisto työpöydältä löytyvästä ikonista. Avaa valmiiksi tehty malli OPCKippi.CATProduct, joka löytyy kansioista C:\Documents and Settings\user\Desktop\DelmiaOPCLabra.

Vasemmalla ylhäällä olevassa puussa näkyy Kippauskone CATProduct, josta löytyy kaikki siihen kuuluvat osat, anturit, sekä liikkumiseen tarvittavat mekanismit ja sisäiset logiikat.

Avaa CSM Device Control Connection workbench: Start - Logic Design - CSM Device Control Connection workbench. Lisää Execution environment: Add Execution environment ja valitse puusta OPCKippi.

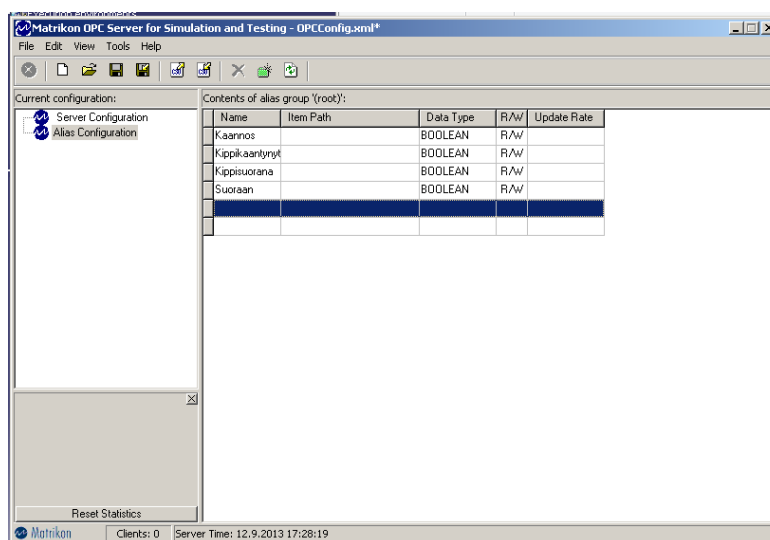
Lisää kippauskone: Add direct devices, sekä PLC OPC-yhteydellä: Add external control. Anturit Kippisuorana ja Kippikääntynyt lisätään klikkaamalla niitä erikseen.



Hyväksy OK:lla ja laitteet ilmestyvät Devices kohtaan.

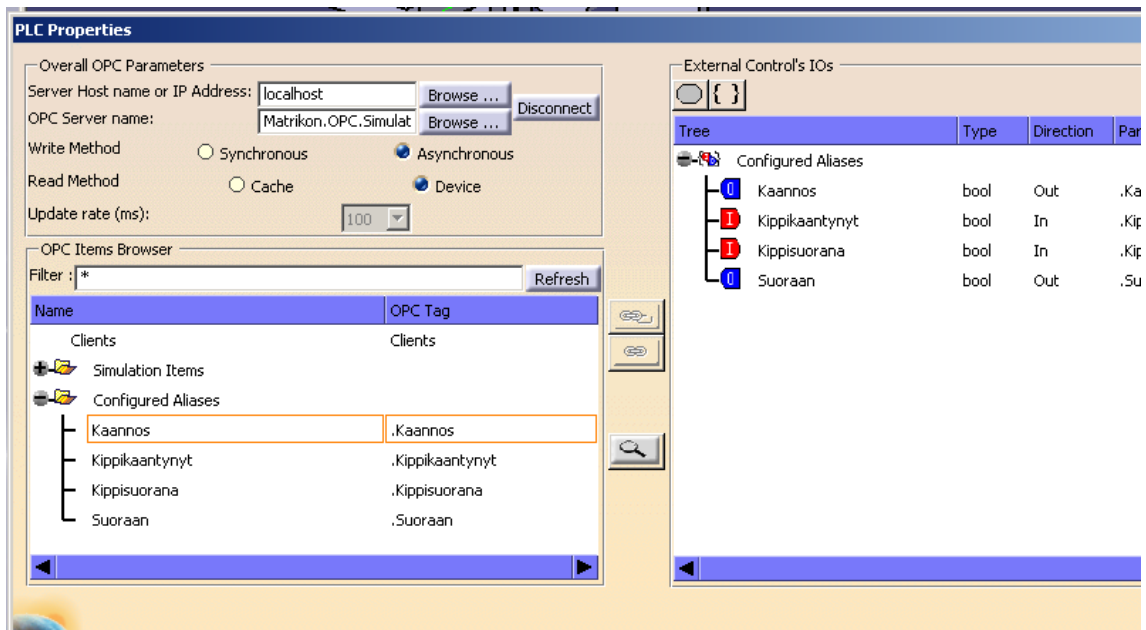
## 2.2. MatrikonOPC

Avaa MatrikonOPC Server for Simulation ohjelma työpöydältä. Lisää Kippikaantynyt, Kippisuorana, Kaannos ja Suoraan Alias Configurationiin.

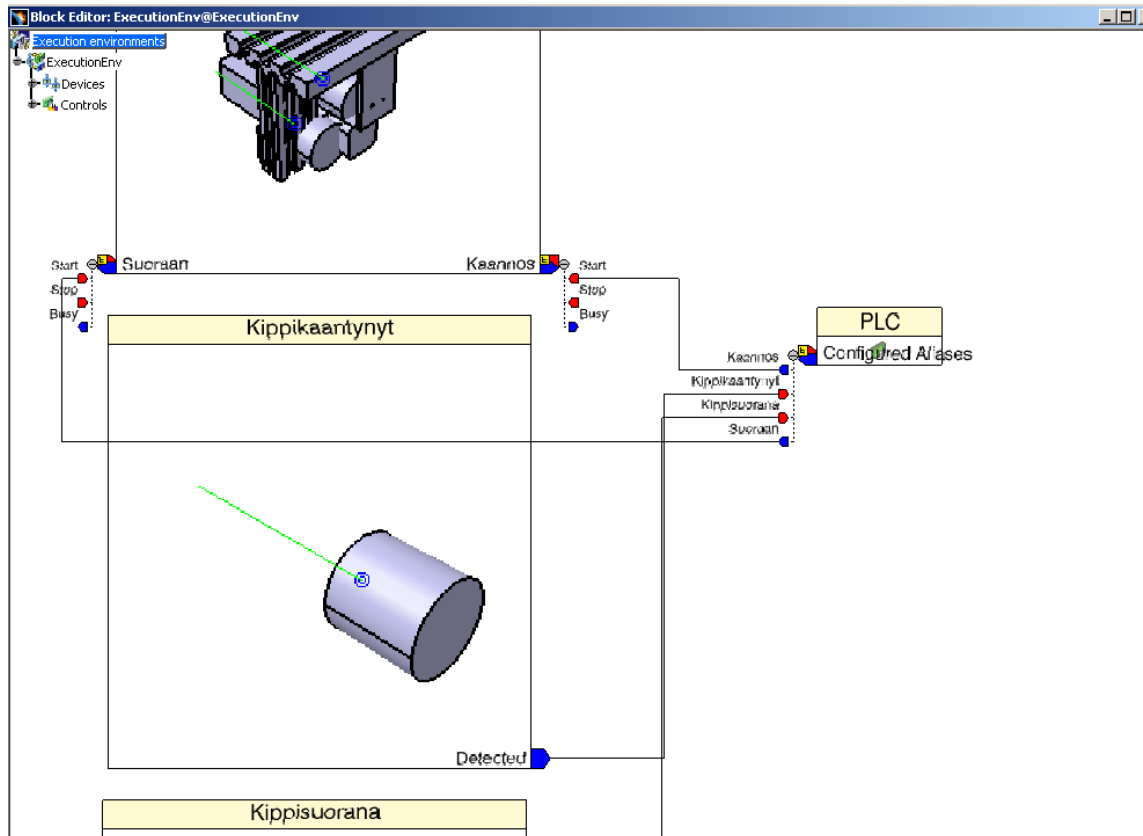


### 2.3 Delmia Simulointi

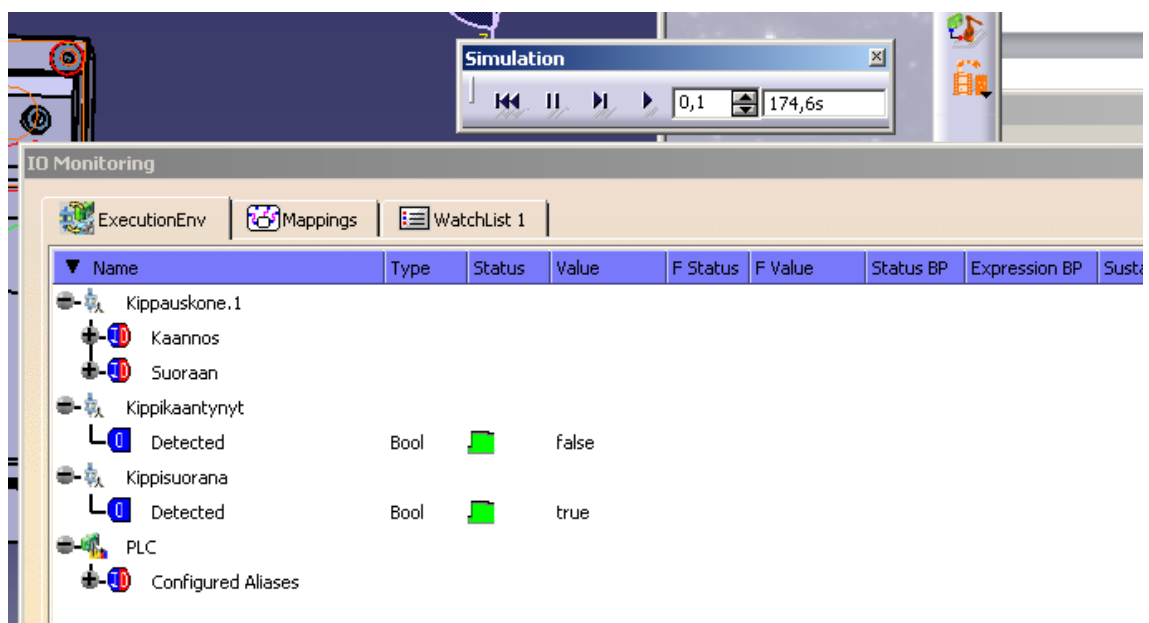
Nyt yhdistetään Delmia OPC:hen. Valitse Execution enviroments - Controls - PLC. Avautuvassa PLC Properties ikkunassa valitse OPC Serveriksi Matrikon.OPC.Simulation.1 ja klikkaa connect. Lisää OPC tagit oikealle External Control's IOs listaan ja valitse näille joko input tai output.



Linkitä laitteet ja OPC portit klikkaamalla Open Connection Editor työkalua ja ExecutionEnv. Vedä Add Connection työkalulla yhteydet oikeisiin kohtiin ja sulje ikkuna.



Nyt valitsemalla Execute ja Execution enviroments päästään simuloimaan systeemiä. Laita simulaatio run tilaan. Kippiä pystytään ohjaamaan kirjoittamalla PLC:n output Kaannos tai Suoraan kohtaan F Valuen arvoksi true.



## **Laboratorioharjoitus**

Delmia Process

Metropolia Automaatiotekniikka



## 1. Johdanto

Tässä työssä on tarkoituksena tutustua Delmia V5R19 ohjelmaan sillä tehtävän simulaatioharjoituksen avulla. Delmia on Catia V5-käyttöliittymän päälle rakennettu digitaaliseen valmistukseen ja prosessisimulointiin erikoistunut ohjelmistokokonaisuus.

Työssä on valmiiksi tehty malli automaatiolaboratoriosta löytyvästä linjaston osasta. Malliin on tehty valmiiksi alkuosa prosessin liikkeistä ja tarkoituksena on tehdä loppuosa samaan tyyliin valmiiksi.

## 2. Tehtävä

### 2.1 Prosessin testaus

Käynnistä Delmia V5R19 ohjelmisto työpöydältä löytyvästä ikonista. Avaa valmiiksi tehty malli Processlabra.CATProcess, joka löytyy kansioista C:\Documents and Settings\user\Desktop\DelmiaProcessLabra.

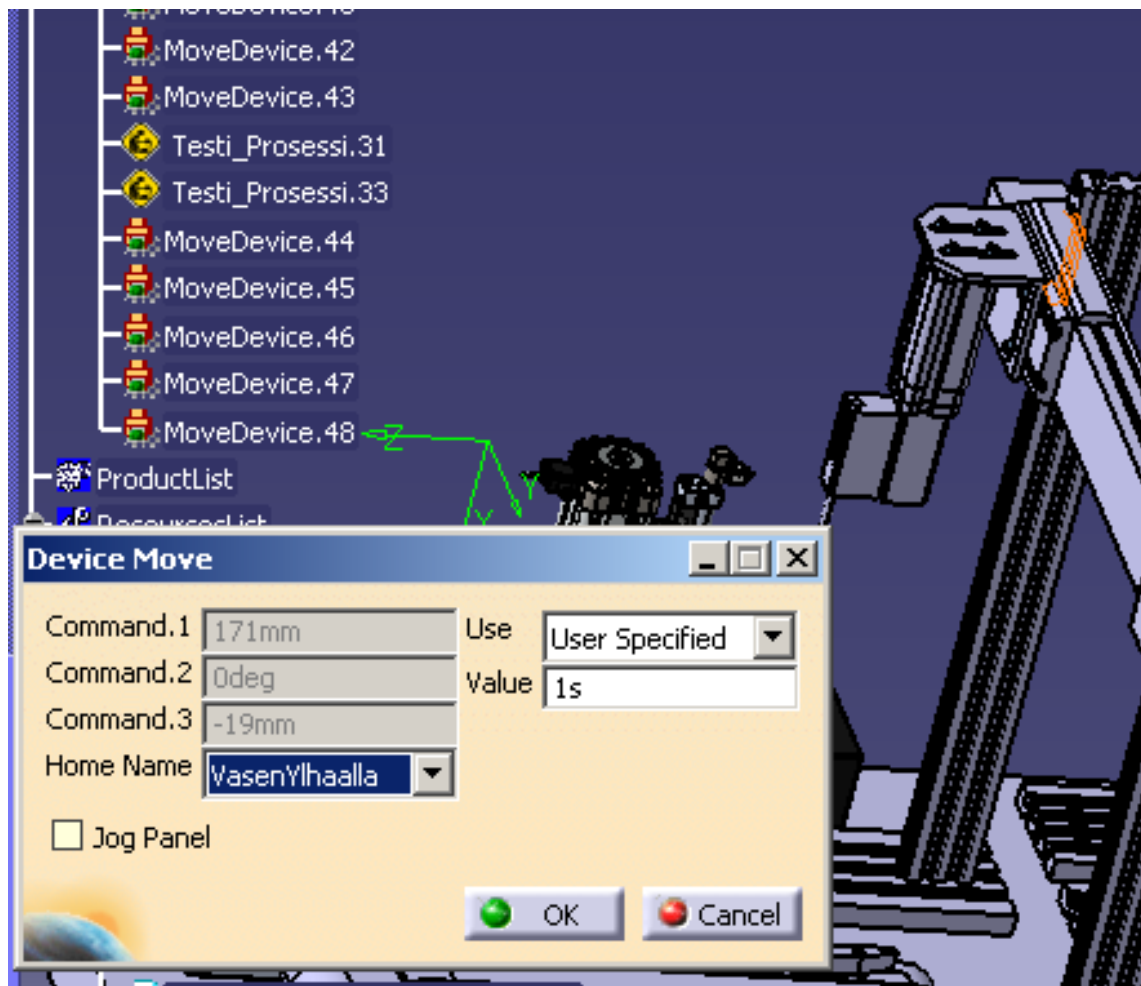
Vasemmalla reunassa olevassa puussa ResourcesListin alla näkyy Product1 CATProduct, josta löytyy linjaston osaan kuuluvat komponentit erillisinä CATprodukteina. Ylempänä puussa löytyy itse prosessi ProcessListin alta.

Alkuosa prosessista on tehty jo valmiiksi. Avaa Workcell Sequencing workbench: Start - Resource Detailing - Workcell Sequencing. Pääset simuloimaan prosessia klikkaamalla ylälaidasta löytyvää Process Simulation nappulaa. Laita simulaatio run tilaan ja testaa nykyistä prosessia.

### 2.2 Prosessin jatkaminen

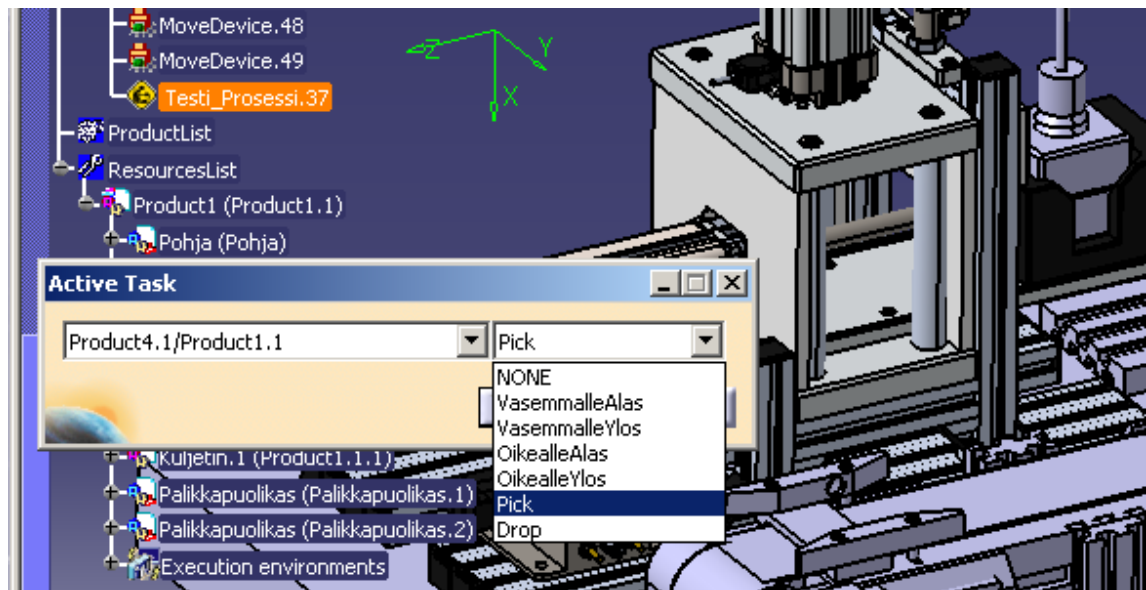
Aja prosessi nykyiseen lopputilanteeseen joko ajamalla simulaatio loppuun, tai valitsemalla puusta viimeinen liike. Seuraavaksi lisätään ohjelmaan palikkojen liikuttaminen prässikoneelta kuljettimelle.

Luodaan sivulle liikkuvan sylinterin liike valitsemalla Creates a Predefined Move Activity ja klikkaamalla viimeisintä MoveDevice kohtaa puusta. Valitaan After the Selected Activity ja valitaan SivulleliikkuvaSyl.1 joko kuvasta tai puusta. Valitaan valmiiksi määritetty kotipiste VasenYlhaalla alaspvetovalikosta.

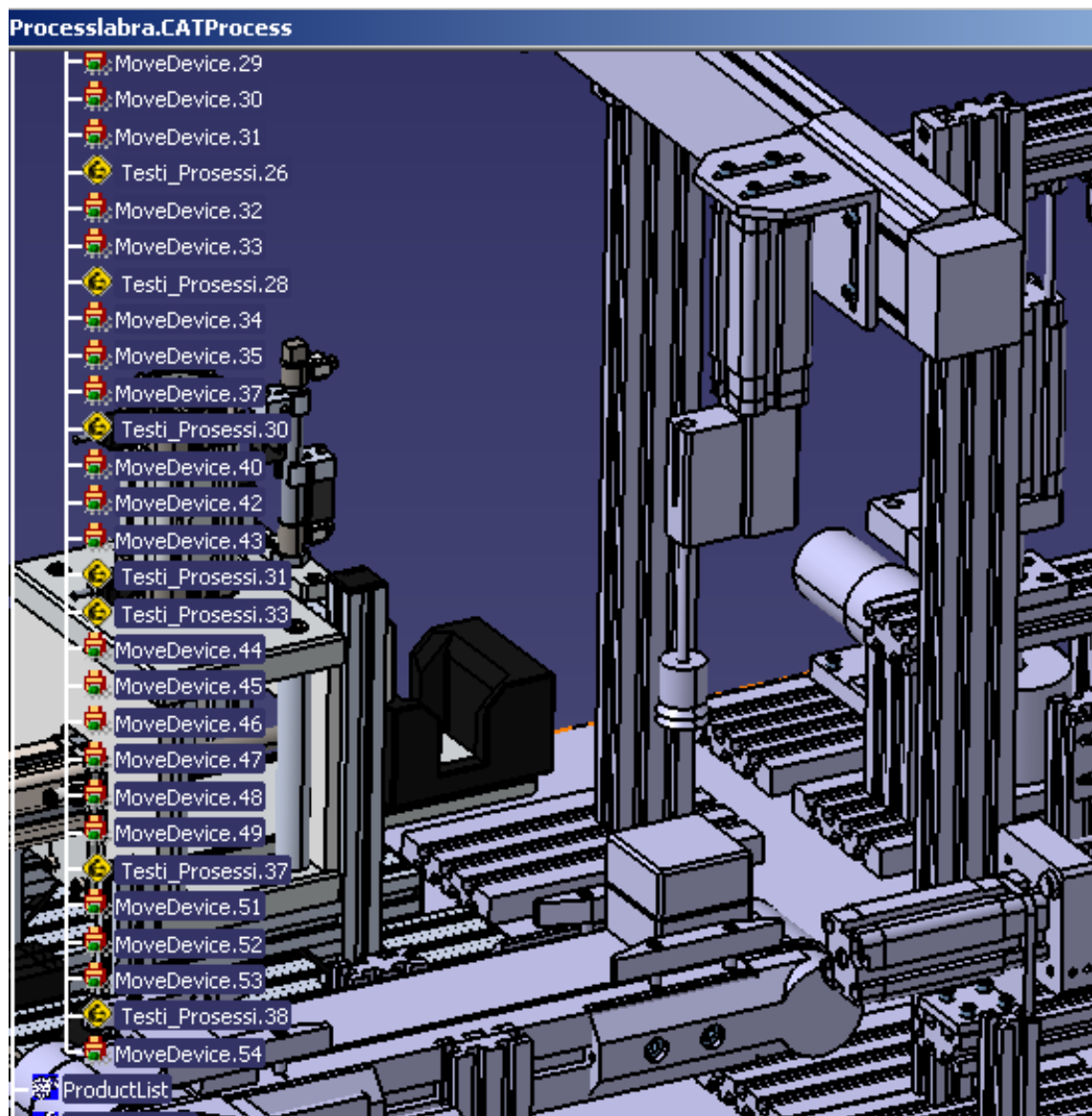


Nyt liike on määritetty prosessilistaan. Seuraavaksi jatketaan lisäämällä samalla tavalla liike kotipisteeseen VasenAlhaalla, joka on ylemmän palikan päällä.

Lisätään palikoihin tarttuminen viimeisimmän liikkeen perään. Insert Activity ja klikataan viimeisintä liikettä listasta, listasta valitaan Testi\_Prosessi ja Add as Successor. Sitten Assign a resource ja valitaan äsken luotu Testi\_Prosessi ja valitaan vielä SivulleliikkuvaSyl1 vasemmalla olevasta puusta. Lisätään vielä pick toiminto: Set an Active Task - Testi\_Prosessi - Alaspvetovalikko Pick.



Lisätään perään loput liikkeet samaan tapaan kuin tehtiin aikasemmin: VasenYlhaalla - OikeaYlhaalla - OikeaAlhaalla - Drop - OikeaYlhaalla. Drop toiminto tehdään samaan tapaan kuin Pick.



Lopputilanne.